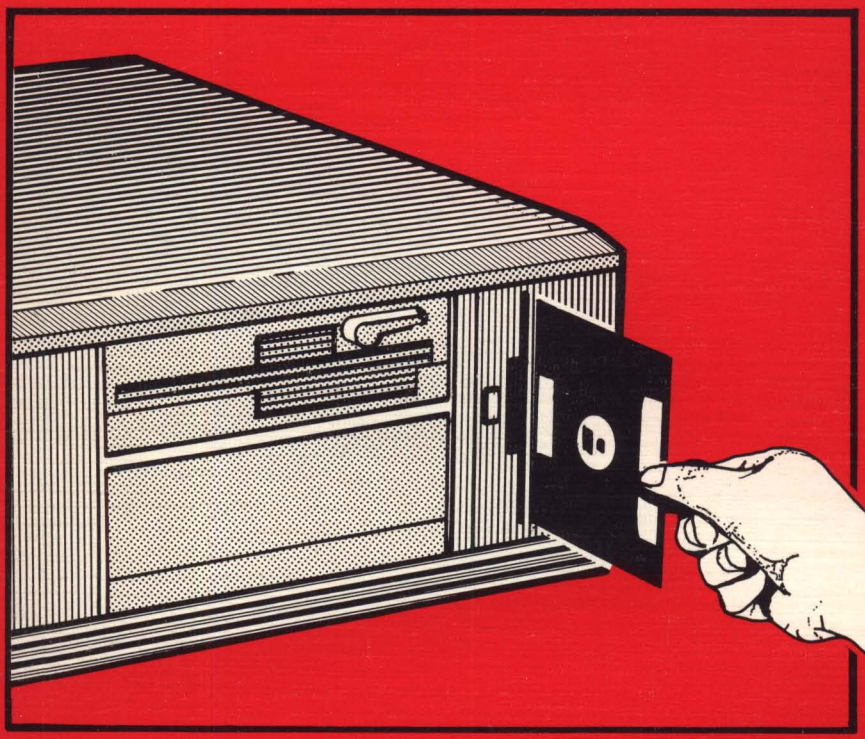
REVISED EDITION
NOW COVERS ALL VERSIONS
3.X AND 4.X

A Concise Introduction to MS-DOS

N. KANTARIS



A Concise Introduction to MS-DOS

ALSO AVAILABLE

(By the same author)

BP232 A Concise Introduction to MS-DOS

BP243 BBC-BASIC86 on Amstrad PCs & IBM Compatibles Book 1 - Language

BP244 BBC-BASIC86 on Amstrad PCs & IBM Compatibles Book 2 - Graphics and Disk Files

BP250 Programming in FORTRAN 77

BP258 Learning to Program in C

BP259 A Concise Introduction to UNIX

BP260 A Concise Introduction to OS/2

BP264 A Concise Advanced User's Guide to MS-DOS

BP274 A Concise Introduction to SuperCalc5

BP270 A Concise Introduction to Symphony

BP279 A Concise Introduction to Excel

BP283 A Concise Introduction to SmartWare II

BP284 Programming in QuickBASIC

BP288 A Concise Introduction to Windows 3.0

BP294 A Concise Introduction to Microsoft Works

BP302 A Concise User's Guide to Lotus 1-2-3 Release 3.1

BP314 A Concise Introduction to Quattro Pro 3.0

A Concise Introduction to MS-DOS

by

Noel Kantaris

BERNARD BABANI (publishing) LTD.
THE GRAMPIANS
SHEPHERDS BUSH ROAD
LONDON W6 7NF
ENGLAND

PLEASE NOTE

Although every care has been taken with the production of this book to ensure that any projects, designs, modifications and/or programs, etc., contained herewith, operate in a correct and safe manner and also that any components specified are normally available in Great Britain, the Publishers and Author(s) do not accept responsibility in any way for the failure (including fault in design) of any project, design, modification or program to work correctly or to cause damage to any equipment that it may be connected to or used in conjunction with, or in respect of any other damage or injury that may be so caused, nor do the Publishers accept responsibility in any way for the failure to obtain specified components.

Notice is also given that if equipment that is still under warranty is modified in any way or used or connected with home-built equipment then that warranty may be void.

© 1987, © 1989 & © 1992 BERNARD BABANI (pub.) LTD

First Published – August 1987
Revised Edition – January 1989
Reprinted – September 1989
Reprinted – March 1990
Reprinted – September 1990
Reprinted – June 1991
Revised Edition – January 1992
Reprinted – April 1992
Reprinted – September 1992
Reprinted – January 1993

British Library Cataloguing in Publication Data:

Kantaris, Noel
A concise introduction to MS-DOS
1. MS-DOS (Computer operating system)
I. Title
005.4'3 qa76.6

ISBN 0 85934 177 1

Printed and Bound in Great Britain by Cox & Wyman Ltd, Reading

ABOUT THIS BOOK

To help the beginner, this concise guide to PC/MS-DOS, has been written with an underlying structure based on 'what you need to know first, appears first'. However, the book is also circular, which means that you don't have to start at the beginning and go to the end. The more experienced user can start from any section, as each section has been designed to be self contained.

This book does not seek to replace the documentation you receive with the PC/MS-DOS operating system, but only to supplement and explain it. The book covers all the latest DOS versions, including the whole range of versions 3.x and 4.x, as implemented by IBM for their microcomputers, and by Microsoft on behalf of manufacturers of other *compatible* personal microcomputers, but excluding DOS version 5.0 which is the subject of a separate book. The book covers systems with both hard disc-based drives, and floppy disc-based drives as applicable to the PC, XT, AT, PS/2 and compatible microcomputers.

A separate section of the book deals with the enhancements to be found in the PC/MS-DOS versions 4.x, one aspect of which is substantially different to earlier versions of the operating system. This is due to the addition of a DOS shell, a menu-driven graphical interface, which makes this version easier to use by newcomers to the world of DOS. The menus employed in the DOS shell are consistent with those used in Microsoft Windows (prior to version 3.0) which, according to the designers of this version, was supposed to make it easier for those who became familiar with it to migrate to the OS/2's Presentation Manager - a more advanced operating system which seeks to replace MS-DOS. However, as OS/2 can only run on PCs based around the 80286, 80386, and higher processors, the millions of users with machines based on the 8086 and 8088 processors will be confined to the MS-DOS operating system and will have to learn their way around it, even though the DOS shell of version 4.x could provide the closest link, be it only an artificial one, between DOS and a future migration to OS/2.

This book was written with the busy person in mind. You don't need to read hundreds of pages to find out most that there is to know about the subject, when a few pages can do the same

thing quite adequately! Naturally, selectivity of subject matter is of paramount importance, so that the reader does not miss vital points. In particular, the book seeks to bring to the forefront and exploit the inherent simplicity in the MS-DOS operating system by presenting, with examples, the principles of what you need to know, when you need to know them.

At the same time, the book has been written in such a way as to also act as a reference guide, long after you have mastered most MS-DOS commands. To this end, a summary of the commands supported by the MS-DOS operating system is given in the penultimate section of this book. The commands are explained with relevant examples and, as such, the section can serve as a quick reference guide.

ABOUT THE AUTHOR

Graduated in Electrical Engineering at Bristol University and after spending three years in the Electronics Industry in London, took up a Tutorship in Physics at the University of Queensland. Research interests in Ionospheric Physics, led to the degrees of M.E. in Electronics and Ph.D. in Physics. On return to the UK, he took up a Post-Doctoral Research Fellowship in Radio Physics at the University of Leicester, and in 1973 a Senior Lectureship in Engineering at The Camborne School of Mines, Cornwall, where since 1978 he has also assumed the responsibility of Head of Computing.

ACKNOWLEDGEMENTS

I would like to thank colleagues at the Camborne School of Mines for the helpful tips and suggestions which assisted me in the writing of this book.

TRADEMARKS

DR-DOS is a registered trademark of Digital Research Corporation

IBM, **PC**, **XT**, **AT**, **PS/2**, **PC-DOS** and **OS/2** are registered trademarks of International Business Machines Corporation

Intel 8086, **8088**, **80286**, **80386** and **80486** are registered trademarks of Intel Corporation

Lotus 1-2-3 is a registered trademark of Lotus Development Corporation

MS-DOS and **Microsoft Windows** are registered trademarks of Microsoft Corporation

Norton Commander and **Q&A** are registered trademarks of Symantec Corporation

PC Tools is a registered trademark of Central Point Software International

Quattro Pro and **Turbo C** are registered trademarks of Borland International Inc

SuperCalc is a registered trademark of Computer Associates

Xtree Gold is a registered trademark of Executive Systems Incorporated.

CONTENTS

1. **INTRODUCTION** 1
   - Versions of MS-DOS 1
   - The Structure of MS-DOS 2
     - Booting up the System 3
   - Internal DOS Commands 5
     - The DATE & TIME Commands 6
   - External DOS Commands 7
   - Files and the Disc Directory 7
     - The DIR Options 9
     - Using Wildcard Characters 10
   - Rules for Entering Commands 11

2. **MANAGING DISC FILES** 13
   - The FORMAT Command 13
     - Type and Size of Discs 14
   - The SYS Command 15
     - Differences Between Disc Drives 15
   - The COPY & DISKCOPY Commands 16
   - The DISKCOMP & COMP Commands 17
   - The DELETE Command 18
   - The RENAME Command 18
   - The CHKDSK Command 19
   - The XCOPY Command 20

3. **THE DIRECTORY TREE** 21
   - Managing Directories 23
   - The Directory Listing 25
     - The Directory Tree 26
   - Renaming Directories 27

4. **THE EDLIN LINE EDITOR** 31
   - The Insert Command on a New File 32
   - The List Command 32
   - The Edit Command 33
   - The Insert Command on an Existing File 33
   - The Delete Command 34
   - The Move & Copy Commands 34
   - The Search Command 34
   - The Search & Replace Command 35

The Transfer Command 36
Exiting Edlin 36

5. SYSTEM CONFIGURATION 37
The CONFIG.SYS File 37
Configuration Commands 38
The AUTOEXEC.BAT File 42
The PATH Command 43
Simple Batch Files 44
The APPEND Command 45
The % Replacement Parameter 46
Special Batch-file Commands 47
Combining Batch Files 49
The COMMAND.COM Processor 52

6. MANAGING YOUR SYSTEM 53
Changing the Access Date of a File 53
The SET Command 53
The TYPE Command 53
The VER & VOL Commands 54
The MORE & SORT Filter Commands 55
The PRINT & PRTSC Commands 56
The BACKUP & RESTORE Commands 56

7. MS-DOS VERSION 4.x 59
Installing MS-DOS 4.x 60
The MS-DOS Shell 61
The File System 64
Changing the Directory & Logged Drive 64
The File Sub-menu 65
The Options Sub-menu 66
The Arrange Sub-menu 66

8. COMMAND SUMMARY 69

9. GLOSSARY OF TERMS 83

INDEX 91

1. INTRODUCTION

Most 16-bit microcomputers use Microsoft's Disc Operating System (MS-DOS) as the prime means of interaction between user and computer. Owners of IBM PCs know this operating system as PC-DOS or DOS, which is IBM's implementation of MS-DOS. The name MS-DOS will be used throughout this book to distinguish this operating system from another popular one, namely Digital Research's DR-DOS.

Versions of MS-DOS

Since its inception in 1981, MS-DOS has been the standard operating system for personal computers and by now is being used by more than 12 million people. As the number of users increased over the years, so too has the complexity of applications run on their PCs. To meet these ever increasing demands, MS-DOS has also increased its functionality several times in the form of *new* versions, as shown in the table below.

Version	*Date*	*Main changes in functionality*
1.0	1981	Original Disc Operating System
1.25	1982	Support for double-sided discs
2.0	1983	Support for sub-directories
2.25	1983	Support for extended character set
3.0	1984	Support for 1.2MB floppy disc and larger capacity hard disc
3.1	1984	Support for PC networks
3.2	1986	Support for 3½" floppy disc
3.3	1987	Support for PS/2 computer range
4.0	1988	Support for extended memory (EMS), hard disc partitions beyond 32MB and a graphical interface DOS shell
4.01	1989	Fix bugs in 4.0 version.

One aspect of version 4.x differs from earlier versions. This is the due to the addition of a DOS shell - a menu-driven graphical interface - which makes this version easier to use for the newcomer to the DOS environment. The menus employed in the DOS shell are consistent with those used in versions of Microsoft Windows prior to version 3.0. The additional functions and advantages gained by using the DOS shell are discussed in a separate section of this book (Chapter 7).

It is assumed here that the reader is familiar with handling floppy discs, floppy and hard disc drives, and that the installation manual which comes with every microcomputer has been read and complied with. What this book tries to do is supplement the MS-DOS operating manual by explaining the various commands with ample working examples which is something that these manuals seldom seem to do; they are excellent for reference purposes, for those who already know, but learning from them is almost impossible.

The MS-DOS operating system consists of a collection of small, specialised programs that make up the working environment which allows you to create and save programs, copy or delete data files from disc or perform other input and output (I/O) operations, such as finding a program or a file on a particular disc or printing the contents of that file on the printer. In general, MS-DOS is the mirco's administrator and understanding the way it works is very important. Running a computer without understanding its operating system is similar to trying to run a library without any knowledge of librarianship. Very soon chaos will be the order of the day.

At first sight, the various commands within MS-DOS might appear difficult to understand and remember, particularly if the first encounter with them is made through the operating manual which was designed, after all, to act as a reference manual and was never intended to be used to learn from it. However, there is a certain simplicity in the way MS-DOS likes to receive instructions and once this simplicity is understood, using its various commands becomes natural.

This book seeks to bring to the forefront, and exploit, this inherent simplicity in MS-DOS by presenting, with examples, the principles of what you need to know, when you need to know them. However, in Chapter 9 you will find a summary of *all* the MS-DOS commands, explained with appropriate and relevant examples, which will serve as a quick reference guide long after you have mastered most of the MS-DOS commands.

The Structure of MS-DOS

To understand how to use MS-DOS you must understand its underlying structure. The various MS-DOS administrative functions are contained in three, separate, main files (later on, we will explain what files mean and their naming convention).

These are:

MSDOS.SYS
IO.SYS
COMMAND.COM

or IBMDOS.COM, IBMBIO.COM and COMMAND.COM, in the case of the IBM PC.

The first file is the core of the operating system, while the second one, also called the Basic Input Output System (BIOS), allows the core to communicate with the hardware. It is the BIOS that is adapted by manufacturers of different hardware so that the operating system can appear to function in the same way, even though there might be differences in hardware design. The last file, COMMAND.COM, is the Command Processor which analyzes what is typed at the keyboard, and if correct, finds and starts execution of the appropriate command.

MS-DOS has nearly thirty built-in commands, normally referred to as 'internal commands', instantly available to the user as they reside in memory. In addition to these internal commands, there are over sixty 'external' commands which are to be found on the System and Utility discs supplied by the manufacturer of the operating system. The machine program which makes up each of these external commands is saved in a *file* under an appropriate name with a .COM or .EXE extension to the filename (more about this later). Collectively, these internal and external commands make up the computer's Disc Operating System (DOS). These commands will be examined in detail in the following sections of this book.

Booting up the System:

To start up the computer, usually referred to as booting up the system, the MS-DOS disc which contains the System files (known as the System disc), must be in the A: drive for a floppy disc-based system, or the files must have been transferred on the C: drive for a hard disc-based system. It is assumed here that you have followed the manufacturer's instructions on how to format the hard disc and transfer the System and all other files from the System and Utility discs onto it. If this is not the case, then do so before going on any further.

In short, when you receive your new computer, you might need to run a utility called FDISK (Fixed Disc) which allows

preparation of the hard disc to run DOS or any other operating system, if this was not done already by your dealer. The utility allows you to partition the hard disc in such a way as to allow more than one operating system to reside on it, or to partition your large physical hard disc to several smaller logical discs. Normally, you'll be using the hard disc to run MS-DOS only, on the C: partition. Be very careful with this utility, because proceeding any further with it will format your hard disc automatically, without any warning, with consequent loss of all data that might reside on it. Nonetheless, on formatting the hard disc, the utility transfers onto it the two hidden system files and copies the COMMAND.COM file onto it, but does not copy the rest of the MS-DOS operating system files. To achieve this, and assuming that the MS-DOS system disc is still in the A: drive, type

```
A>COPY A:*.* C:
```

repeating the command with the second (utility) disc, if there is one, inserted in the A: drive. However, before you start using the FDISK utility, make sure that your computer dealer has not actually done all this for you, in which case FDISK is best left alone. Also, before using this utility, make absolutely sure that you have read and complied with your manufacturer's instructions on the subject.

If, after successfully booting up the system (indicated by the appearance of a prompt A> or C>, for booting from a floppy or hard disc drive, respectively), you were to type DIR, followed by pressing the <Enter> key, to find out the contents of the disc, neither of the first two System files (MSDOS.SYS and IO.SYS) would appear on the directory list as they are hidden so that you can not delete them. Only the third file (COMMAND.COM) would be displayed.

In addition to these three special files, there are a number of other files supplied by MS-DOS which perform various important tasks. These files are collectively known as the MS-DOS utilities and will be examined in detail later. To be able to distinguish between disc drives, MS-DOS refers to them by a letter followed by a colon, i.e. A: or C: for the prime drive of the appropriate system. In a twin floppy disc-based system, there are two drives; A: and B:, with drive A: being the leftmost or uppermost of the two, while on a hard disc-based system there

is a floppy disc drive, A: and a hard disc drive, C:. However, DOS allows you to also refer to drive A: as drive B:, so that you can copy files from one floppy disc to another using a single floppy disc drive. Users on networked systems can access a network hard disc by assigning it as another drive on their micro, namely as E: or Z:

On booting up the micro from a MS-DOS System disc, the following tasks are performed:

- A self test on its Random Access Memory (RAM) is performed
- A check is made to see if a floppy disc is in drive A:, and if there is, whether it is a System disc. If it is, it boots the system from the A: drive
- If no floppy exists in drive A:, an attempt is made to boot the system from drive C:, if there is one, otherwise in the case of the IBM, it goes into Read Only Memory (ROM) based BASIC
- Configures the system by executing the CONFIG.SYS file
- Reads the BIOS and the MS-DOS operating system
- Loads into RAM the COMMAND.COM file so that internal commands can be made available instantly
- Executes the commands within the AUTOEXEC.BAT file, if one exists, otherwise it asks for the Date and Time which can be reset at this point. Pressing the <Enter> key, confirms what is displayed.

Should you receive any error message while these tasks are being performed, you could restart the process, after rectifying the error, by pressing simultaneously the three keys marked **Ctrl**, **Alt** and **Del**, shown as **Ctrl+Alt+Del** in the rest of the book. This will re-boot the system.

Internal DOS Commands

MS-DOS has nearly thirty internal commands built into it which are instantly available as they reside in memory. These are listed on the next page.

Command	*Meaning*
BREAK	Sets the Ctrl+Break check on or off
CALL	Calls one batch file from another
CHCP	Displays or changes the active code page
CD or CHDIR	Change the current directory
CLS	Clears the screen
COPY	Copies files
CTTY	Changes the standard Input/Output device
DATE	Displays or sets the system date
DEL	Deletes the specified files
DIR	Displays the disc directory
ECHO	Sets Echo to on or off
EXIT	Exits to the previous command level
FOR	Repeats a command for each item in a set
GOTO	Jumps to a labelled line within the same batch file
IF	Allows conditional execution of commands within a batch file
MD or MKDIR	Makes (creates) a new directory
PATH	Searches alternative directories
PAUSE	Pauses execution of the batch file
PROMPT	Changes the system prompt
REM	Allows remarks to be added in a batch file
REN or RENAME	Renames files
RD or RMDIR	Removes (deletes) a directory
SET	Changes the system parameters
SHIFT	Allows more than 10 replaceable parameters in a batch file.
TIME	Displays or sets the system time
TYPE	Displays a specified text file
VER	Displays the MS-DOS version
VERIFY	Checks disc writing
VOL	Displays the disc volume label

The internal DOS commands (most of which will be explained later), together with the rest of the operating system, occupy some 40 Kbytes of RAM, as they are loaded into memory on booting up the system. As an illustration, consider the two commands which allow you to re-set the date and time of your computer's internal clock.

The DATE & TIME Commands:

Your computer is equipped with an internal clock, whose date and time can be changed. Typing the command

```
A>DATE
```

at the A> or C> prompt, evokes the response

```
Current date is dd/mm/yy
Enter new date:
```

at which point you can either type a new date or press <Enter> to indicate that date is not to be changed. The above date format assumes that you have included the command COUNTRY=xxx (or equivalent in the case of PC-DOS 3.3), where xxx is a three digit code representing your country, in your CONFIG.SYS file (to be discussed later), otherwise the date will be shown in mm/dd/yy format.

Typing the command

```
A>TIME
```

at the prompt, evokes the response

```
Current time is Hrs:Mins:Secs
Enter new time:
```

at which point you can either type a new time or press <Enter> to indicate that time is not to be changed.

External DOS Commands

DOS provides over sixty additional commands which, to avoid eating up more of the computer's memory, reside on the System disc. These are known as external commands and can only be invoked if that disc is inserted in either the A: or B: drive (for a floppy disc-based system). For a hard disc-based system, these additional commands would have been transferred onto the C: drive and can, therefore, be accessed directly from that drive.

A complete list of all the MS-DOS commands is given in Chapter 8.

Files and the Disc Directory

To see what files are held on the System disc, you must access the disc directory by typing DIR at the A> prompt, as follows:

```
A>DIR      (and press the <Enter> key)
```

The A> prompt will be displayed on booting up the system if you are using a floppy disc-based system, otherwise the prompt will be C>. To change the C> prompt to A> prompt, and vice versa,

requires you to change the logged disc drive by typing A: at the C> prompt, or C: at the A> prompt. Now, typing DIR against the appropriate prompt, displays a directory listing. Amongst the many files to be listed will be the ones shown below, the size and creation dates of which are dependent on the version of MS-DOS you are running on your computer. The ones shown below are related to MS-DOS version 4.01.

Filename	*Extension*	*Size*	*Date*	*Time*
ANSI	SYS	9178	10-04-89	9:00a
APPEND	EXE	11154	10-04-89	9:00a
BACKUP	COM	36958	10-06-89	9:00a
CHKDSK	COM	17755	10-06-89	9:00a
COMMAND	COM	37513	10-04-89	9:00a
CONFIG	SYS	252	10-04-89	9:00a
COUNTRY	SYS	12806	10-04-89	9:00a
DISKCOPY	COM	11488	10-04-89	9:00a
EDLIN	COM	14069	10-04-89	9:00a
FORMAT	COM	23981	30-05-90	9:00a
KEYBOARD	SYS	23312	10-08-89	9:00a
LABEL	COM	4458	10-04-89	9:00a
PRINT	COM	14339	10-04-89	9:00a
RESTORE	COM	37134	23-06-89	9:00a
SYS	EXE	11616	10-04-89	9:00a
VDISK	SYS	7998	27-09-89	9:00a
XCOPY	EXE	17055	10-04-89	9:00a

Note that a filename consists of up to 8 alphanumeric characters (letters and numbers only) and has a three letter extension, separated from the filename by a period, i.e. COMMAND.COM or CONFIG.SYS, without any spaces in between, unlike the listing appearing on your screen, where the periods have been omitted and the extensions have been tabulated as above. Some of these files might have different extensions from the ones shown above, i.e. BACKUP.COM might appear as BACKUP.EXE in your system, as the extensions tend to differ for different versions of MS-DOS. The size of each file (in bytes) is also given on the listing together with the date and time it was created, which again might differ for different versions.

The extensions .COM, .SYS and .EXE are the most common extensions of the files which make up MS-DOS. They contain

instructions which are executed directly by the computer. Other extensions commonly used by programs or users are:

.BAK .BAS .BAT .DAT .DOC TXT .TMP

which indicate 'back-up' files, 'Basic' programs, 'batch' files, 'data' files, 'document' files, 'text' files and 'temporary' files, respectively.

The DIR Options:

Returning to the result of issuing the DIR command; what is more likely to have happened in your case is that the listing of the first half of the files on your disc will have scrolled out of view. In all, there are approximately forty utility files on the System disc and you can only see the last twenty or so. To stop the scrolling of a long directory, use the /P option after the DIR command, as follows:

```
A>DIR/P
```

which will page the directory, displaying twenty files at a time. Alternatively, you could see all these files on your screen by using the /W option, as follows:

```
A>DIR/W
```

which lists the files sideways, as follows:

ANSI	SYS	APPEND	EXE	ATTRIB	EXE
BACKUP	COM	CHKDSK	COM	COMMAND	COM
CONFIG	SYS	COUNTRY	SYS	DISKCOPY	COM
EDLIN	COM	FORMAT	COM	MORE	COM
PRINT	COM	RESTORE	COM	SORT	EXE
SYS	EXE	VDISK	SYS		

Note that in this case the information relating to the size of each file and the date and time of its creation has been omitted from the listing.

The slash (/) options in MS-DOS commands, like the /P and /W in the DIR command above, are also referred to as switches. However, as this name could be confused with physical switches, the name 'options' will be used throughout.

Using Wildcard Characters:
You can limit the information which appears on your screen by being more selective with the use of wildcards. For example, to list all the .EXE files on your disc, type

```
A>DIR *.EXE
```

where the wildcard character '*' stands for 'all' files.

Note that spaces are very important to MS-DOS. Had you not included a space after DIR in the above command, MS-DOS would have responded with its favoured error message,

```
Bad Command or file name
```

which does not tell you very much, except that MS-DOS does not understand you!

The wildcard character '*' can also be used as part of the filename. For example,

```
A>DIR BAS*.*
```

will list all the files with all extensions on the logged drive (in this case drive A:), starting with the three characters BAS, irrespective of the ending of the filenames.

The full MS-DOS command should also specify which drive you want to access, but can be omitted if the command refers to the currently logged drive. Thus,

`A>DIR BAS*.*`	or
`A>DIR A:BAS*.*`	will access the specified files on drive A:, while
`A>DIR B:BAS*.*`	will access the specified files on drive B:.

Alternatively, you can change the logged drive by simply typing its identification letter at the prompt. For example,

`A>B:`	will change the logged drive, indicated by changing the prompt, to

```
B>_
```

which indicates that the currently logged drive is now B:. All further commands which do not specify a different drive, will access drive B:. To revert back to the previously logged drive, type A: at the B> prompt.

A more precise wildcard is the query character '?' which can be substituted for a single character in a filename. For example, assuming that there are several consecutively numbered files on your disc with filenames TEXT1.DOC to TEXT999.DOC, typing

```
A>DIR TEXT?.DOC
```

will list all files with the extension .DOC, from TEXT1 to TEXT9, but not those within the range TEXT10 to TEXT999. On the other hand, using two consecutive query characters in the filename, such as

```
A>DIR TEXT??.DOC
```

will list all files with the extension .DOC, from TEXT1 to TEXT99, but exclude those within the range TEXT100 to TEXT999.

To list all the files from TEXT1 to TEXT999 you must use the wildcard character '*' in place of the single query, as follows:

```
A>DIR TEXT*.DOC
```

Finally, typing

```
A>DIR *.*
```

will display all files with all extensions which, of course, has the same effect as typing

```
A>DIR
```

Nevertheless, the *.* is worth noting as it is the most useful three-character combination in MS-DOS and will be mainly used in housekeeping commands to be explained later.

Should you ever want to find out whether a particular file exists on a disc, just type its name after the DIR command. If the file exists, MS-DOS will display it, otherwise the message

```
File not found
```

will appear on your screen.

Rules for Entering Commands

Commands can be entered in either uppercase or lower-case letters, but you must provide a space between the command and its parameters. For example, to obtain a listing of all the

.EXE files on the floppy in the A: drive in wide format, you can either type

```
A>DIR *.EXE /W
```

or

```
A>dir *.exe /w
```

but you must type one space between the R (or r) and the asterisk (*). The space between the E (or e) and the slash sign (/) is optional; its presence only serves to improve redability.

If DOS informs you that you have made a mistake while typing a command - you will soon know this because on pressing the <Enter> key, DOS will reply with

```
Bad command or file name
```

you can edit what you have typed previously by pressing the right arrow key which causes DOS to display the command a letter at a time. Keep pressing the right arrow key until you have reached the offending part of the command which can be over-typed. If need be, you can add extra letters to the command by pressing the <Ins> key, or remove extra letters by pressing the <Del> key, then you can use the right arrow key again to continue revealing the rest of the command.

checking existing disk format.
Formatting 1.44M
"Invalid media or Track 0 Bad
- Disk unusable."
Format terminated.

2. MANAGING DISC FILES

MS-DOS provides several commands which help you to manage your disc files efficiently. Some of these commands are internal and some are external. If the commands under discussion are external commands, it will be pointed out so you can insert the System disc in the logged drive which is the drive indicated by letter on the screen prompt.

The FORMAT Command

One of the first things you will need to do, as a new user, is to make a working copy of your System disc, or favoured software package, or just a back-up copy of your programs or data. Such packages and/or data are far too valuable in terms of money or time invested in producing them to be used continually without the safeguard of back-up copies.

Again, it is assumed that in the case of a hard disc-based system, your hard disc has already been formatted according to your manufacturer's instructions when setting up the system, and that all the MS-DOS external command files have been transferred onto it.

A new floppy disc must be formatted before it can be used by your computer's operating system. A floppy disc that has been formatted in one computer, can only be used in another computer if they are compatible and use the same operating system.

To format a disc, in the case of a hard disc-based system where the logged drive will be drive C:, insert the new floppy disc in the A: drive and type.

```
C>FORMAT A:/S/V
```

In the case of a twin-floppy based system, insert the System disc in the A: drive, as FORMAT is an external command and needs to be loaded into RAM from the System disc, then insert the new floppy disc in the B: drive and type

```
A>FORMAT B:/S/V
```

Drive C: (or A: in the case of a floppy disc) is accessed momentarily, the FORMAT utility file is loaded into RAM and executed. You are then given instructions to insert a floppy disc

in drive A: (or B: in the case of a floppy disc system), and press <Enter> to begin.

Be very careful never to format an already formatted disc (particularly the C: drive), as *all* files that might be on it will be lost.

The two options, typed after the slash character (/), have the following meaning:

The /S option instructs MS-DOS to copy the two hidden system files and the COMMAND.COM file onto the newly formatted disc. This will be required if you intend to use the disc to boot up the system.

The /V option allows you to give a Volume label to your new disc, after formatting is completed.

Type and Size of Discs:

There are some additional options that can be used with the FORMAT command which, however, are dependent on the type of disc drive being used and size of disc. These are as follows:

Disc type	*Disc size*	*Parameters*
160/180 KB	5.25"	/1, /8, /4
320/360 KB	5.25"	/1, /8, /4
1.2 MB	5.25"	/4, /N, /T
720 KB/1.44 MB	3.5"	/N, /T

where

/1 formats only one side of the disc.

/8 formats 8 sectors per track.

/4 formats 40 tracks with 9 sectors per track for 360 Kbytes using a 1.2 Mbyte high-capacity disc drive. This option must be used if you are using double-density and not high-capacity, double-density discs in a 1.2 Mbyte drive.

/N specifies the number of sectors per track to format, written as /N:9 for nine sectors.

/T specifies the number of tracks, written as /T:40 for forty tracks. To format a 720 Kbytes double-sided disc in a high-capacity 3.5" disc drive (1.44 Mbytes), use options /N:9/T:80.

If options /N or /T are specified, then both parameters must be entered. All other options can be used separately or omitted altogether from the command. Omitting the /S option from the FORMAT command saves disc space.

The SYS Command

Should you change your mind after you have formatted a disc without the use of the /S option, you can use the external SYS command to transfer the System files from the logged disc drive onto a previously formatted disc, inserted in another drive. The command takes the form:

`C>SYS A:` in the case of a hard disc-based system, or

`A>SYS B:` in the case of a floppy disc-based system.

To successfully transfer the operating system to a disc with this method, the disc must either be newly formatted or else have space on it for the transfer of the operating system by perhaps already having a different version of it on the target disc.

Finally, note that the SYS command transfers only the two hidden files of the operating system which means that you must use the COPY command (see next section) to transfer the COMMAND.COM and CONFIG.SYS files.

Differences Between Disc Drives:

The PC, XT and compatibles have 360 Kbyte double-sided, double-density disc drives. Discs are formatted with 40 tracks per side, 9 sectors per track with 0.5 Kbyte of information per sector, resulting in 360 Kbyte capacity.

The IBM AT, XT286 and compatibles have 1.2 Mbytes high-capacity double-sided disc drives. Discs are formatted with 80 tracks per side, 15 sectors per track with 0.5 Kbyte of information per sector resulting in 1.2 Mbytes capacity. However, each track takes the same physical space as that of the 360 Kbyte drive, the difference being that the 1.2 Mbyte drive writes tracks that are half the width of the 360 Kbyte drive.

Discs formatted on 1.2 Mbyte disc drives with the /4 option use only one half of the width of each of the 40 tracks. This information can easily be read by a 360 Kbyte drive (as a result of tolerance in signal level), provided the other half of the track is completely clear. Should you now use the 360 Kbyte drive to write to the disc, information is written to the full width of the

track which can still be read by the 1.2 Mbyte disc drive (again, as a result of tolerance in signal level). However, any subsequent writing to such a disc using the 1.2 Mbyte drive, results in changes to only one half of the track width. The result is half a track containing the new information with the corresponding other half of the same track containing the old, half-overwritten information, which makes it impossible for the 360 Kbyte disc drive to make any sense of it. Thus, to avoid such incompatibilities between disc drives, always format double-density discs in a 360 Kbyte drive. Attempting to format a double-density disc to 1.2 Mbytes will result in many bad sectors with future loss of data becoming highly possible.

There are no such problems arising from the use of 3.5" discs which have been formatted as 720 Kbytes in a high-capacity (1.44 Mbytes) disc drive and subsequently used to read or write to them by either a 720 Kbyte or a 1.44 Mbyte disc drive.

Again, avoid formatting non high-density discs to 1.44 Mbytes to avoid future loss of data. On the other hand, if you do format accidentally a high-capacity disc to 720 Kbytes (this can happen if you use the DISKCOPY command - see below - to copy files from a 720 Kbytes disc to a high-density disc), do not attempt to re-format the high-density disc to 1.44Mbytes, because the higher current used by the disc drive to format it to 720 Kbytes can not be wiped out by the re-formatting process.

The COPY & DISKCOPY Commands

To copy all files on the disc in the logged drive to the disc in the A: drive, type

```
C>COPY *.* A:
```

Note the most useful three-character combination in MS-DOS, namely *.* which means 'all filenames with all extensions'.

However, if you wanted to copy a set of files from drive A: to drive C:, while being logged onto the C: drive, type

```
C>COPY A:*.DOC C:
```

which means: COPY from the A: drive all the files with extension .DOC to the C: drive.

The /V option can be used at the end of the COPY command to force MS-DOS to verify that the file(s) it has copied can be read. For example,

```
C>COPY FORMAT.COM A:/V
```

will copy the formatting utility file FORMAT.COM from the logged drive to the A: drive and force verification that the file can be read.

Both the formatting and copying can be done in one go by using the DISKCOPY command, as follows:

```
C>DISKCOPY A: B:
```

which will copy all the files from the A: drive, to the B: drive and format the disk in the B: drive at the same time, if not already formatted.

However, if the disc in the B: drive is already formatted, but at different capacity to that in the A: drive, then the disc in the B: drive is re-formatted to the same capacity as that in the A: drive.

You can not use the DISKCOPY command to copy files from a floppy disc to a hard disc, or to copy files between two floppy discs of different size (such as 3½" and 5¼" discs).

Note: It is preferable to use the FORMAT and COPY commands than use the DISKCOPY command when copying all files from one disc to another. The reason is that bad sectors are frozen out when formatting a disc with the FORMAT command and the subsequent use of COPY, avoids these sectors. The DISKCOPY command on the other hand, seeks to make an identical copy (sector by sector) of the original disc which means that it attempts to write on bad sectors, if any, which might lead to an unsuccessful copy operation.

The DISKCOMP & COMP Commands

These two external utilities are mostly needed if you use the DISKCOPY command. The first one compares the contents of two discs, while the second one compares the contents of two files. The commands take the following form:

`C>DISKCOMP A: B:`	compares the discs in the A: and the B: drives, while
`C>COMP A:MORE.COM`	compares the MORE.COM file to be found in the C: and the A: drives.

The DELETE Command

Unwanted files on a disc can be deleted, as follows:

`C>DEL EXAMPLE.TMP`	deletes EXAMPLE.TMP on the C: drive
`C>DEL A:EXAMPLE.TMP`	deletes EXAMPLE.TMP on the A: drive
`C>DEL *.*`	deletes all files on the logged drive.

Luckily, the use of the DEL *.* command evokes the response

```
Are you sure? (Y/N)
```

which acts as a safety net. It is a good idea to always check what you are about to delete from your disc by first using the DIR command.

For example, say you intend to delete all the .TMP files from your disc. First use DIR *.TMP and if what is displayed on screen is what you want to delete, then type DEL and press the **F3** function key. This has the effect of displaying on the screen the last command you typed on the keyboard, minus the characters you typed prior to pressing the **F3** key. Thus, DEL replaces DIR and the use of **F3** displays the rest of the command. In this way you avoid making any mistakes by re-typing.

As you will, no doubt, use this command at some time or another, it will be prudent to copy both your AUTOEXEC.BAT and CONFIG.SYS files into your DOS sub-directory (see next chapter). In this way you will have copies of these vital files away from harm's reach!

The RENAME Command

The REN command is used to rename files. As an example, let us assume that we want to rename a file on the disc in the logged drive from its current filename OLDFILE.DOC to the new filename NEWFILE.DOC. This can be done as follows:

```
C>REN OLDFILE.DOC NEWFILE.DOC
```

Note the importance of spaces after REN and in between the two file names. The command can be interpreted as:

```
Rename from filename1 to filename2
```

To rename a file on a disc in a disc drive other than the logged drive, the disc drive specification must also be included in the command, as follows:

```
C>REN A:OLDFILE.DOC NEWFILE.DOC
```

Note that, if you intend to rename a file and give it a filename that already exists on disc, you must first delete the unwanted file before renaming, otherwise MS-DOS will refuse to obey your command, informing you that the filename you have chosen already exists on disc.

The CHKDSK Command

This command checks a disc, reporting whether it is formatted as single or double sided, how many files are stored on the disc, how much space they take and how much space is still available. The command also checks the RAM, reporting on both the total memory available and the number of bytes still free. The command takes the form:

`C>CHKDSK`	which checks the disc in the logged drive, or
`C>CHKDSK A:`	which checks the disc in the A: drive.

The additional /F option, allows CHKDSK to also do some routine maintenance, namely fixing lost clusters. A cluster is the minimum amount of space (one or more sectors) that can be allocated to a file on disc.

Each disc has a file-allocation table (FAT) where a note is kept of which clusters have been allocated to which file. However, with heavy disc use, the file-allocation table can be corrupted and using CHKDSK will report 'lost clusters found'. The /F option, converts these into files and gives them the general name FILExxxx.CHK, where xxxx starts with 0000 and increments by 1. These files can then be checked and perhaps deleted if found to be useless.

The XCOPY Command

The XCOPY command allows us to copy files and directories, including lower level sub-directories, if they exist (see following section for a full explanation of these), to the specified destination drive and directory. The command takes the following form:

C>XCOPY source_filespec destination [options]

where *source_filespec* specifies the source file or drive and directory you want to copy and *destination* can be the drive to which you want this source file to be copied to. For example,

```
C>XCOPY A:*.* B:
```

will copy all files in the A: drive to the B: drive. If you only have one floppy drive, you will be prompted to change discs.

Some of the *options* available (for a full list see the 'Command Summary' section) are as follows:

/D copies source files which were modified on or after a specified date

/P prompts the user with '(Y/N?)' before copying files

/S copies directories and their sub-directories unless they are empty

/V causes verification of each file as it is written

XCOPY copies all files and sub-directories in the specified directory from one disc to another by reading from the source disc as many files as possible into memory, then copying them to the target disc. This is unlike the COPY command which copies each file in turn, therefore, taking much longer. However, when the target disc becomes full, XCOPY stops and does not ask for another disc to be inserted in the target drive. XCOPY can copy files even when the two discs are of different format, unlike the DISKCOPY command which requires that the source and target disc be the same format.

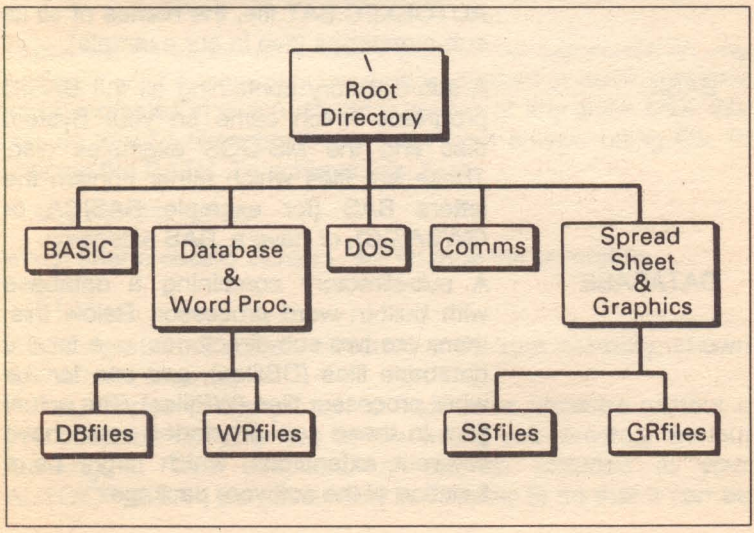
If you do not specify a path in the source filespec, XCOPY starts from the current directory. If you specify only a drive or path, but no files, XCOPY assumes you mean 'all files'. You can specify a target filename that is different from the source filename which causes files to be renamed while copying.

3. THE DIRECTORY TREE

If you are using a system with normal capacity disc drives, then organizing the files you keep on discs is relatively straightforward. The usual method would be to keep similar applications on the same disc, so that one disc might contain files on word processing, another on spread sheets, and another on databases. MS-DOS keeps track of all such files by allocating space on each disc, called a directory, in which such information as the name of each file, its size, the date it was last amended, etc., is kept.

However, as you move up to systems with high-capacity disc drives (1.2 or 1.44 Mbyte floppies) and especially to systems with hard discs of 20, 40 or more Mbytes, the amount of information you can store on them increases so much, that unless you organize the way you keep your files on such discs, you could easily spend all of your time trying to find one.

MS-DOS can help you to organize your files on disc by providing a system of directories and sub-directories. The key to the MS-DOS system is the 'root' directory, indicated by the back-slash sign (\), which is the main directory under which a number of sub-directories can be created. In turn, each sub-directory can have its own sub-directories, as shown below.



The root directory is shown here with five sub-directories under it, while two of these have their own sub-directories below them. For maximum efficiency, the root directory should contain only the System and start up files, together with information on its sub-directories, a kind of an index drawer to a filing system.

Files in different sub-directories can have the same name because MS-DOS can be told which is which via a system of PATH names. For example, a file in the SSFILES sub-directory could have the same name, say SALARY.TMP, as one in the GRFILES sub-directory. Nevertheless, we can instruct MS-DOS to fetch the file in the SSFILES sub-directory by giving its path name which is:

```
\SPREADSH\SSFILES\SALARY.TMP
```

whereas that of the file in the GRFILES sub-directory is:

```
\SPREADSH\GRFILES\SALARY.TMP
```

In the example shown previously, the contents of the various sub-directories might be as follows:

\	The root directory, containing the two hidden System files MSDOS.SYS and IO.SYS, the Command Processor COMMAND.COM, the CONFIG.SYS file, the AUTOEXEC.BAT file, the names of all its sub-directories (five in our example).
BASIC	A sub-directory containing all the BASIC programs which came on your System disc and the MS-DOS examples disc. These are files which either contain the letters BAS (for example BASICA or GWBASIC), or have a .BAS extension.
DATABASE	A sub-directory containing a database with built-in word processor. Below this, there are two sub-directories; one for the database files (DBfiles), and one for the word processor files (WPfiles). The actual files in these sub-directories could have different extensions, which might be a function of the software package.

DOS	A sub-directory containing all the MS-DOS files comprising the external MS-DOS commands.
COMMS	A sub-directory containing communications programs, propriety back-up software, etc.
SPREADSH	A sub-directory containing an integrated spreadsheet and graphics package. Below this, there are two sub-directories, one for the spread sheet files (SSfiles), and one for the graphics files (GRfiles). Again, the actual files in these could have different extensions which might be a function of the software package.

Managing Directories

MS-DOS provides three special commands for the creation and management of sub-directories. these are:

Command	*Meaning*	*Example*
MD	Make sub-directory	C>MD \BASIC
CD	Change directory	C>CD \BASIC
RD	Remove directory	C>RD \BASIC

These will be explained in detail shortly, but before we go any further, it will be extremely useful and prudent to have a prompt which indicates in which directory we are at any given time. We can do this by changing the prompt from always being C>, to indicate the PATH. To achieve this, type

```
C>PROMPT $P$G
```

which after pressing <Enter>, will change the prompt to

```
C:\>_
```

indicating that the current directory is the root directory, shown by the back-slash (\).

This change of the prompt is imperative because without it you could be copying files to the wrong sub-directory without realising it. This command should be included in your AUTOEXEC.BAT file (discussed in Chapter 5) so that it can be

executed on booting up the system. To do so, however, you will need to learn to use the **edlin** line editor which is explained in the next chapter. For the time being, to benefit from this change of the prompt, just type the command at the C> prompt.

Before a directory can be used, it must exist. If it does not, you can make it with the MD command. To make the sub-directory called BASIC, so that you could transfer to it all BASIC programs and files from your MS-DOS System and examples discs, type the following line

```
C:\>MD\BASIC
C:\>_
```

which makes the BASIC sub-directory in the root directory and waits for further commands. The full path was given after the MD command, by first specifying the root directory with the use of the back-slash (\), followed by the name of the sub-directory.

To transfer files from a disc in the A: drive, first change directory using the CD command by typing

```
C:\>CD\BASIC
C:\BASIC>_
```

which causes the prompt to change, indicating that MS-DOS has actually changed directory. Without the prompt change, you would have had the typical 'where am I?' problem. Note that the moment we create a sub-directory we tend to refer to its parent as directory, even though itself might be a sub-directory to another parent directory.

To copy all BASIC programs and files to this directory, place in turn each MS-DOS disc in the A: drive and type

```
C:\BASIC>COPY A:BAS*.*
C:\BASIC>COPY A:*BAS*.*
C:\BASIC>COPY A:*.BAS
C:\BASIC>_
```

Alternatively, we could have issued these commands from the root directory without first changing directories. As an example, the first line of the three copy commands given above, would have to be typed as

```
C:\>COPY A:BAS*.* C:\BASIC
```

which, however, involves more typing on your part.

The Directory Listing

Were you to create the above directories with their sub-directories on your hard disc (you could, of course, set-up these on a floppy disc to avoid changing the configuration of your hard disc) and then type the command DIR, you would see the following display:

```
 Volume in drive C is MS-DOS_401
 Volume Serial Number is 1832-12E4
 Directory of  C:\

COMMAND  COM     37513 10-04-89   9:00a
AUTOEXEC BAT       327 18-05-91   9:00a
CONFIG   SYS       252 06-04-91   9:00a
BASIC        <DIR>     05-01-92  10:00a
DATABASE     <DIR>     05-01-92  10:00a
DOS          <DIR>     05-01-92  10:00a
COMMS        <DIR>     05-01-92  10:00a
SPREADSH     <DIR>     05-01-92  10:00a
       8 File(s)   20238592 bytes free

C>
```

The volume name (given above as MS-DOS_401) might be different in you case, as it depends on the label you gave your disc just after formatting. If you did not give a disc a label, then that line will read 'Volume in drive C has no label'. The second line of the display depends on your system and you have no control over it.

Note that directories are distinguished from files by the inclusion of the letters <DIR> against their name, and the order of their appearance depends on the order of their creation.

These days, many program packages create their own directory structure during installation and suggest names for the required directories and sub-directories into which they deposit their files. What they don't do, is to create sub-directories for your data. These you must create yourself in order to avoid adding your data files into the same directory as the one holding the program files. Doing so, makes it easier for you to make a back-up of your data files onto floppy discs, and lays the foundation for a structured hard-disc system.

The Directory Tree:
MS-DOS contains a very useful external command called 'tree'. Its use allows you to see pictorially the way DOS structures directories and sub-directories, as shown below. The command has been enhanced considerably in v4.x and its use actually draws schematically the directory structure. In previous MS/PC-DOS versions the use of this command tended to produce only a character listing of sub-directories under their appropriate directories.

```
C>tree

Directory PATH listing for Volume MS-DOS_401
C:.
├──BASIC
├──DATABASE
│   ├──DBFILES
│   └──WPFILES
├──DOS
├──COMMS
└──SPREADSH
    ├──SSFILES
    └──GRFILES

C>
```

The topmost level in a disc directory structure is the root directory which created automatically by MS-DOS when you format a new disc and, unlike other directories, it does not have a unique name. It is within the root directory that you can create additional directories and sub-directories and give them unique names.

When you create directories and sub-directories, MS-DOS creates what is known as the dot (.) and double dot (..) entries. These shorthand notations can be used to identify the current and the parent directory, respectively. Thus, typing

```
DIR ..
```

lists the files in the parent directory of the current sub-directory, while typing

```
CD ..
```

changes the logged directory from the current sub-directory to the parent directory.

Renaming Directories

Should you be dissatisfied with the name of an existing directory and you want to rename it, you will have to create another directory giving it your preferred name, copy to the newly created directory all the files from the unwanted directory, delete all files from the unwanted directory, and then remove the unwanted directory from its parent directory. This procedure is essential because:

(a) you can not rename directories, and
(b) you can not remove directories unless they are empty.

As an example of the above procedure, let as assume that we have created, as discussed previously, a sub-directory to the root directory, called DATABASE. To have created such a sub-directory, we would have had to return to the root directory from whichever sub-directory we were at the time, by typing

```
CD\
```

at the prompt.

We now proceed to create a sub-directory to the DATABASE directory, called DOCBASE, by first changing directory from the root directory to that of DATABASE, as follows:

```
C:\>CD\DATABASE
C:\DATABASE>_
```

then make a sub-directory called DOCBASE by typing

```
C:\DATABASE>MD DOCBASE
```

at the prompt. Note that we have omitted the back-slash from in front of the sub-directory name which causes it to be made in the currently logged directory. Had we included the back-slash, the sub-directory DOCBASE would have been created as a sub-directory of the root directory.

Alternatively, we could make DOCBASE without first changing directory by issuing the MD command from the root directory, but giving the full path specification, as follows:

```
C:\>MD\DATABASE\DOCBASE
```

Having made sub-directory DOCBASE, copy into it your files from the A: drive, as discussed previously.
Let us now assume that for some reason the directory name DOCBASE offends you and you would like to change it to WPBASE instead. To do this you will have to type in the following commands, assuming you are at the root directory:

```
C:\>CD \DATABASE
C:\DATABASE>MD WPBASE
C:\DATABASE>CD \DATABASE\WPBASE
C:\DATABASE\WPBASE>COPY \DATABASE\DOCBASE\*.*
C:\DATABASE\WPBASE>CD \DATABASE\DOCBASE
C:\DATABASE\DOCBASE>DEL *.*
Are you sure? (Y/N)Y
C:\DATABASE\DOCBASE>CD \DATABASE
C:\DATABASE>RD DOCBASE
C:\DATABASE>_
```

It is not necessary to include a space after the MD, CD, or RD directory commands when typing them in. The space was included above only to make it easier for you to read.

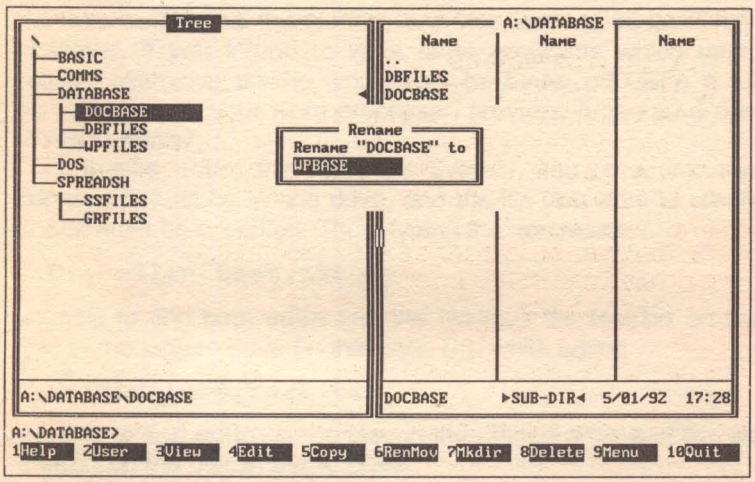
In order of appearance, these lines of commands do the following:

(a) change directory to DATABASE
(b) make a sub-directory called WPBASE
(c) change directory to WPBASE
(d) copy from sub-directory DOCBASE all files to the logged sub-directory
(e) change directory to DOCBASE
(f) delete all files from logged directory
(g) MS-DOS asks for confirmation
(h) change directory to DATABASE
(i) remove sub-directory DOCBASE.

As you can see, this is cumbersome, so think how you want to structure your hard disc before plunging into it blindly.

Re-structuring directories and sub-directories, moving files from one directory to another, or making back-ups of groups of files, can be made easy for users of MS-DOS v4.x with the DOSSHELL program (see Chapter 7). Users of lower numbered versions of DOS can use certain proprietary software, such as Symantec's Norton Commander, Central Point's PC Tools, or Executive Systems' Xtree Gold, to name but a few.

Such proprietary software disc utilities are extremely easy to use, and to illustrate the point, the rename display from Norton Commander is shown below:



The program is similar to DOSSHELL in the way that it displays the directory tree on the left-half of the screen, with the files residing within the highlighted directory being listed on the right-half of the screen. In Norton Commander, the important function keys, and what they do, are displayed on the last line of the screen. Thus, to rename the DOCBASE sub-directory of DATABASE from DOCBASE to WPBASE, you simply highlight the DOCBASE sub-directory in the directory tree, press the **F6** function key, and type the new name in the dialogue box that appears in the middle of the screen.

It is, perhaps, worth casting your mind back to what was required of you in order to do the same job when using straight MS-DOS commands and compare it with what was just done with the help of one of these utilities!

It is obvious, therefore, that such proprietary disc utilities can save you quite a bit of effort and frustration and are well worth the initial outlay, in both money and the time taken to master their use. You must, of course, bear in mind when thinking of purchasing one of these utilities that they can do much more than simply rename directories.

4. THE EDLIN LINE EDITOR

MS-DOS provides you with a simple line editor, called EDLIN, and you should become familiar with its use. In general, **edlin** allows the creation and editing of ASCII files. These are text files which when sent to the screen or printer are interpreted as text, unlike the .COM or .EXE files which are binary.

Edlin can also be used to create the source code of various programming languages, such as Fortran and C. In such cases, remember to give the file the appropriate extension, which for the two languages mentioned, are **.for** and **.c**, respectively. However, if you intend to write large programs which might require extensive editing, you might be better off using a full screen editor or your word processor, provided it can save files in ASCII format.

To invoke **edlin**, the MS-DOS System disc or a disc that contains it must be in one drive, and the file you want to create or edit must be specified. Thus, typing the command:

```
C:\>edlin test.txt
```

expects to find both **edlin** and the fictitious file **test.txt** on the disc in the logged drive (in this case C:), while typing

```
C:\>edlin A:test.txt
```

expects to find **edlin** on the disc in the logged drive and the file **test.txt** on the disc in the A: drive.

If the file does not exist on the specified disc, then **edlin** responds with

```
New File
*_
```

and waits for further commands, while if the file already exists, then **edlin** loads the file into RAM and responds with

```
End of input file
*_
```

Note the '*' prompt which is characteristic of **edlin**.

Let us now create a text file, called **test.txt**, which we will use to demonstrate the power of **edlin**. To start, type at the MS-DOS prompt

```
C:\>edlin test.txt
```

which should cause **edlin** to respond with

```
New File
*_
```

if that file does not exist on your disc. If it does exist and you do not want to spoil its contents, then type **q** (for quit) and press the <Enter> key.

The Insert Command on a New File

To insert lines of text, use the command **i** (for insert) at the prompt. In the case of a new file, as no lines of text exist in the file, type 1i and then type in the short text given below.

```
*1i
       1:*first line of text
       2:*second line of text
       3:*^C
*_
```

After typing 1i at the prompt, **edlin** responds by giving a new line number (in this case 1:) with an asterisk after it to indicate that this is the current line. At this point we type 'first line of text'. On pressing the <Enter> key, **edlin** gives us an additional line number, now 2:*, into which we type 'second line of text'. Again, on pressing <Enter>, we are offered a further line number, and so on. To end the insertion mode, type **Ctrl+C**. The character **^C** is the two-key depression **Ctrl+C** (hold the key marked **Ctrl** down and press the **C** key).

The List Command

To see what text is in the file, type **1** (for list) at the prompt, as follows:

```
*1
       1: first line of text
       2:*second line of text
*_
```

The line numbers are inserted by **edlin** so that you can refer to the line you want to edit. The '*' in line 2 indicates that this line was the last to be edited or inserted when **edlin** was used last.

Note that now there is only one current line. Should the file you are listing be very long, listing in this manner causes the current line to appear in the middle of the listing.

To list specific lines, use the **l** command with line numbers. For example,

```
*5,15l
```

will list lines from 5 to 15 inclusive. Note the syntax of the command which is: 'From line number to line number Command'. There must be no comma between the second line number and the command letter.

The Edit Mode

To change the current line, type the new line number and press <Enter>. This puts you in edit mode and will cause the line whose number you typed to be displayed. Pressing <Enter> again, confirms that you are happy with the contents of that line, otherwise you can either press the right cursor key to reveal each letter of that line, or re-type the entire line, making any necessary changes. In our case, we want to change line 2 to

```
second line of text, edited
```

so enter the edit mode and change the line appropriately. This is best done by using the right arrow cursor key to reveal the whole of the existing line and then typing the extra information at the end of it. The **Ins** and **Del** keys can also be used to edit the text.

The Insert Command on an Existing File

To insert lines of text, use the command **i** (for insert) at the prompt. However, be warned. Using **i** by its own will insert the new line before the current line (the one with the * after the line number). To insert lines at any other point, give the line number before the command.

In our case, we would like to insert two additional lines after the existing two. To do this, type

```
*3i
      3:*third line of text
      4:*fourth line of text
      5:*^C
*_
```

Again, insertion mode is terminated in line 5: by pressing **Ctrl+C**. If we now list the contents of the file, we get:

```
*l
       1: first line of text
       2: second line of text, edited
       3: third line of text
       4:*fourth line of text
 *_
```

The last line to be inserted becomes the current line.

The Delete Command

To delete unwanted lines of text, use the **d** command (for delete) at the prompt. However, if you use the **d** command without any number associated with it, you will delete the current line (the one with the asterisk). Therefore, if you want to delete line 13, say, type

```
*13d
```

or if you want to delete a group of lines, type

```
*13,15d
```

which is translated as 'lines 13 to 15 to be deleted'.

The Move & Copy Commands

To move or copy text, use the **m** or **c** commands (for move or copy). These commands must be preceded by three numbers, as follows:

```
*13,15,8m
```

which is interpreted as 'lines 13 to 15 to be moved to a position before line 8'.

Similarly, the **c** command will copy a block and insert it before the given line. To move or copy a single line, the first two numbers in the command will have to be the same. After moving or copying lines, always use the list command to force re-numbering of the file's contents.

The Search Command

To search for the occurrence of a word or a specified number of characters in a file you have created using **edlin**, use the

search command. Just as in the list and delete commands, a line range is first specified, followed by the **s** (for search) command. Thus, typing

```
*1,4s edited
```

evokes the response

```
        2: second line of text, edited
*_
```

which displays the line containing the word 'edited'.

Note that the space between the command **s** and the word 'edited' becomes part of the search string. Had we been searching for the characters 'con' within the word 'second', we would have had to omit the space between the command s and the string 'con'.

The search command finds only the first occurrence of the specified string. To continue the search for further occurrences of the same string, simply type **s** again. Thus, typing

```
*1,4sir
        1: first line of text
*s
        3: third line of text
*_
```

causes **edlin** to first find the string 'ir' in the word 'first' of line 1:, then by typing **s** again, it forces **edlin** to find the same string 'ir' in the word 'third' of line 3:.

The Search & Replace Command

This command is similar to the search command, except that it requires a replacement string. Thus, typing

```
*1,4r edited^Z re-edited
```

will cause all occurrences of the word 'edited' to be replaced by the word 're-edited' in all the specified lines of text. Here, of course, it only occurs once in line 2: of the text. The character **^Z** is the two-key depression **Ctrl+Z** (hold the key marked **Ctrl** down and press the **Z** key), which acts as a delimiter between the two strings. Again note that the space in front of both words becomes part of both the searching and the replacing strings.

The Transfer Command

This command transfers the contents of a file into the file currently being edited. The format of the command is:

[*n*] T *filespec*

where

n specifies the line number where the new data is to be inserted. The data is inserted before the specified line. If the line number is omitted, then the current line is used.

filespec specifies the file that you want to insert the contents of into the current file in memory.

Exiting Edlin

To end the current session and exit **edlin** at any point, type

```
*e
```

which saves a new file under the chosen filename.

However, if the filename already existed on disc prior to using **edlin**, ending **edlin** has the following effect:

First the name of the old file on the disc is given the extension **.bak**, then the new file you have created by editing the old one is saved with the original extension. In this way you can make mistakes without disastrous effects since the system makes a back-up file of the original. If need be, you could delete the **.txt** file and then rename the back-up file (**.bak**) to its original name and extension.

Note that **edlin** is disciplined not to allow editing of back-up files so, should you want to start using **edlin** to edit the contents of a **.bak** file, you must first rename it, by giving it a different extension, before proceeding.

If, on the other hand, you realised that too many mistakes were made during editing, you could use the **q** command to quit, as follows:

```
*q
```

instead of using the **e** command as discussed above. Doing this causes **edlin** to ask you whether you want to abort. Typing **y** (for yes), leaves the name and contents of the original file on disc unaltered.

5. SYSTEM CONFIGURATION

The CONFIG.SYS File

This file allows you to configure your computer to your needs, as commands held in it are executed during booting up the system. The easiest way to create or amend this system file is with the use of the line editor **edlin**, as discussed in the previous section.

Do remember that if you are setting up your system for the first time, you will need to create/change the CONFIG.SYS file that exists on your System disc. If your system has already been implemented by, say, your computer staff, do not edit this file or use **edlin** to look at its contents (use instead the **type** command), unless you have to and you know precisely what you are doing, as the file contains entries that MS-DOS uses to define specific operating attributes.

To use **edlin** to change the CONFIG.SYS file, type

```
C:\>edlin CONFIG.SYS
```

which will cause **edlin** to respond (if both the line editor, **edlin** and the CONFIG.SYS file exist on the disc in the logged drive) by

```
End of input file
*_
```

To see what text is in the file, type **1** (for list) at the prompt, which will display the contents of the file as follows:

```
*1
       1: FILES=8
       2:*BUFFERS=2
*_
```

The meaning of each of these statements is explained below. Now edit each line in succession by typing its number and changing it appropriately as given below, then enter the insert mode, by typing 3i, to add the two additional lines. The final contents of the file should be:

```
FILES=15
BUFFERS=25
```

```
BREAK=ON
COUNTRY=044
```

To end the insert mode, press **Ctrl+C** on the 5: line and then exit **edlin** by typing **e**. This will save your changed file as CONFIG.SYS and the contents of the old file as CONFIG.BAK. Do refer below for the meaning of each and every statement you have been asked to change or add in your CONFIG.SYS file.

Following is a list of the commands that you can include within the CONFIG.SYS file which MS-DOS supports. However, do remember that any changes made to this file only take effect after re-booting which can be achieved by pressing the three keys **Ctrl+Alt+Del** simultaneously.

Configuration Commands:

A brief explanation of the configuration commands, which can be included within the CONFIG.SYS file, is given below:

BREAK — By including the command BREAK=ON in the CONFIG.SYS file, you can use the key combination **Ctrl+C** (hold the key marked Ctrl down and press C) or **Ctrl+Break**, to interrupt MS-DOS I/O functions.

BUFFERS — MS-DOS allocates memory space in RAM, called buffers, to store whole sectors of data being read from disc. The default number of buffers is 2, each of 512 bytes of RAM. If more data are required, MS-DOS first searches the buffers before searching the disc, which speeds up operations. The number of buffers can be changed by using:

```
BUFFERS=n
```

where n can be a number from 1 to 99. However, as each buffer requires an additional 0.5 Kbyte of RAM, the number you should use is dependent on the relative size between the package you are using and your computer's RAM. Best results are obtained by choosing between 10-30 buffers.

COUNTRY MS-DOS displays dates according to the US format which is month/day/year. To change this to day/month/year, use the command

```
COUNTRY=044
```

where 044 is for U.K. users.

Non U.K. users can substitute their international telephone country code for the 044. The default value is 001, for the USA.

Users of a hard disc with PC-DOS 3.3 should enter this statement as

```
COUNTRY=044,437,C:\DOS\COUNTRY.SYS
```

where 437 is the code page of pre-3.3 versions of DOS and COUNTRY.SYS is to be found in the \DOS sub-directory. In PC-DOS 3.3 the extended IBM character set has been changed to accommodate several versions of it by offering several choices on the characters displayed or printed. If you are a newcomer to DOS then use the multilingual code page 850 in which many of the scientific symbols and box graphic characters have been replaced by international characters. This allows more European languages to be supported. However, using any other than code page 437 might cause problems with some application programs as not all would have adapted to the new codes. For more details regarding the new code page numbers and the requirement of using the DEVICE= statement in the CONFIG.SYS file when the new codes are being used, see your DOS reference guide. No additional DEVICE= statements are required with code page 437.

CODEPAGE This command is to be found in PC/MS-DOS versions 3.3 and later. The table that DOS uses to define a character set is called a code page. Thus, include the command

CODEPAGE=437

where 437 is the code page definition of pre-3.3 versions of DOS.

DEVICE

MS-DOS includes its own standard device drivers which allow communication with your keyboard, screen and discs. However, these drivers can be extended to allow other devices to be connected by specifying them in the CONFIG.SYS file. Example of these are:

DEVICE=ANSI.SYS

which loads alternative screen and keyboard drivers for ANSI support - features of which are required by some commercial software.

DEVICE=MOUSEAnn.SYS

allows the use of specific mouse devices.

DEVICE=VDISK.SYS n

allows you to specify the size n in Kbytes (default 64) of RAM to be used as an extra very fast virtual disc. With computers which have more than 640 Kbytes of RAM, the option /E can be used after n in the command to allocate the specified memory size from the extra area of RAM.

DEVICE=DRIVER.SYS

allows you to connect an external disc drive.

DEVICE=EGA.SYS

provides mouse support for EGA modes.

DEVICE=COMn.SYS

specifies asynchronous drivers for the serial ports, where for n=01 specifies an IBM PC AT COM device, and n=02 specifies an IBM PS/2 COM device.

FCBS

Specifies the number of FCBs that can be opened concurrently. The command takes the following form:

```
FCBS=x,y
```

where x specifies the total number of files by FCBs, from 1 to 255, that can be opened at any one time (the default value being 4), and y specifies the number of opened files (from 1-255) that cannot be closed automatically by MS-DOS if an application tries to open more than x files.

FILES

MS-DOS normally allows 8 files to be opened at a time. However, some software such as relational databases, might require to refer to more files at any given time. To accommodate this, MS-DOS allows you to change this default value by using:

```
FILES=n
```

where n can be a number from 8 to the maximum required by your application which usually is 20, although the maximum allowable is 99.

INSTALL

This command is new to DOS v4.x; it executes a FASTOPEN, KEYB, NLSFUNC, or SHARE command during CONFIG.SYS processing. The command takes the following form:

```
INSTALL=filespec[params]
```

where *params* specifies the optional line to pass to the *filespec* which must be FASTOPEN.EXE, KEYB.EXE, NLSFUNC.EXE or SHARE.EXE.

LASTDRIVE

This command is used if additional drives are to be connected to your system, or you are sharing a hard disc on a network. The command takes the form:

LASTDRIVE=x

where x is a letter from A to Z (default E).

REM — REM followed by any string, allows remarks to be entered in the CONFIG.SYS.

SHELL — Manufacturers of some micros provide a 'front end' or an alternative Command Processor to COMMAND.COM as real-mode command-line processor. To invoke this, the command SHELL must be included within the CONFIG.SYS file. The command takes the form:

SHELL=FRONTEND.COM

where FRONTEND is the name of the alternative Command Processor. The default value of SHELL is COMMAND.COM.

The AUTOEXEC.BAT File

This is a special batch file that MS-DOS looks for during the last stages of booting up and if it exists, the commands held in it will be executed. One such command is the KEYB xx which configures keyboards for the appropriate national standard, with xx indicating the country. For the U.K., the command becomes KEYB UK, and you will need to execute it if your keyboard is marked with the double quotes sign on the 2 key and/or the @ sign over the single quotes key and/or the £ sign over the 3 key. To create the AUTOEXEC.BAT file, use the **edlin** line editor as follows:

```
C:\>edlin AUTOEXEC.BAT
```

which expects to find **edlin** on the disc in the C: drive. If AUTOEXEC.BAT does not exist on this disc, then **edlin** will respond with

```
New File
*_
```

at which point you enter the insert mode, type the command and save the created file, as follows:

```
*1i
      1:*KEYB UK
      2:*^C
*e
C:\>_
```

Users of a hard disc with PC-DOS 3.3 and later should enter the KEYB UK statement as

```
1:*KEYB UK 437 C:\DOS\KEYBOARD.SYS
```

where 437 is the code page of pre-3.3 versions of DOS and KEYBOARD.SYS is to be found in the \DOS sub-directory. As mentioned previously under the COUNTRY section, in PC-DOS 3.3 and later, the extended IBM character set has been changed slightly to accommodate several versions of it by offering several choices on the characters displayed or printed. Each such version is referred to by a specific code page number which defines the character set to be used. If you intend to use any other code page than 437, then you should refer to your DOS reference guide.

There are a lot of other commands that can be included in the AUTOEXEC.BAT file, such as setting the PATH, changing directory or executing a program. Such commands will be discussed in detail in the next section. However, in the mean time remember to re-boot the system in order to activate the AUTOEXEC.BAT file after you have created it, by either using the **Ctrl+Alt+Del** key-stroke or simply typing AUTOEXEC at the C:\> prompt, as changes to AUTOEXEC.BAT do not take effect until you do so.

The PATH Command:

Naturally, we would like to be able to use the MS-DOS external commands from anywhere within the directory tree without having to specify where the commands are kept (in this instance, we have transferred them into the DOS directory). The same could be said for the programs kept in the COMMS directory. This can be achieved by the use of the PATH command, as follows:

```
PATH C:\;C:\DOS;C:\COMMS
```

which should be included in the AUTOEXEC.BAT file. PATH can only find program files, that is executable command files

with the extension .EXE or .COM, or files that DOS recognises as containing such commands, as is the case with .BAT files; for data files you must use the APPEND command as will be explained shortly. Note the repeated reference to the C: drive which allows the path to be correctly set even if the user logs onto a drive other than C:.

By now you must have noticed that every time you boot up the system, the commands within your AUTOEXEC.BAT file are echoed (displayed) onto the screen. To clean up the screen of such echoes, change your AUTOEXEC.BAT file to include:

```
ECHO OFF
CLS
PATH C:\;C:\DOS;C:\COMMS
KEYBUK
PROMPT $P$G
ECHO HELLO ... This is your
VER
```

which has the following effect. First echoing is switched off, but only after executing the ECHO OFF command and thus, to clear the screen of the displayed command, we employ the CLS command (for v3.3 and v4.x users these first two commands could be replaced by @ECHO OFF which eliminates the echoing of the command). Then, the path, keyboard and prompt commands are executed unseen, until echo is re-activated by executing the ECHO command with a trailing message which is displayed on the screen, followed by the version (VER) of your MS-DOS.

Simple Batch Files

To complete the implementation of the hard disc, we need to create a few batch files which we will put in a sub-directory of the root directory, called BATCH. This will help to run the system efficiently. For example, we might require to know the exact name of a DOS or a COMMS command. This can be arranged by creating a batch file for each, to display the corresponding directory whenever the appropriate name is typed. As an example, first create the BATCH sub-directory, using the

```
C:\>MD\BATCH
```

command, then use the **edlin** line editor to create the DOS.BAT file in the BATCH sub-directory, as follows:

```
C:\>edlin \BATCH\DOS.BAT
New File
*1i
       1:*ECHO OFF
       2:*CLS
       3:*CD\DOS
       4:*DIR/P
       5:*CD\
       6:*^C
*e
C:\>_
```

In line 3:, the directory is changed to that of DOS and line 4: causes the contents of the DOS directory to be displayed using the paging (/P) option. Finally, line 6: returns the system back to the root directory.

However, before you can use the DOS.BAT batch file, you must include the sub-directory BATCH within the PATH command of your AUTOEXEC.BAT file. Having done this, and re-booted the system so that the changes in your AUTOEXEC.BAT file take effect, typing DOS displays the DOS directory, while typing any external MS-DOS command, invokes the appropriate command. A similar batch file can be built for displaying the COMMS directory, the only difference being in line 3: of the file, so that the correct directory is accessed and displayed.

The APPEND Command:

Furthermore, it is conceivable that the software packages you will be using in place of our imaginary DATABASE and SPREADSH packages (included as sub-directories to the root directory), require you to type their name in order to activate them. However, some packages also include a second file (most likely a data file which might contain information about the screen display) which is loaded from the first when its name is typed. In such cases you cannot use the PATH command within the AUTOEXEC file to point to the particular package, as MS-DOS will search for the second (data) file in the root directory as its extension will most likely be .SCR or .OVL and DOS will not search for it down the PATH.

To overcome this, users of MS-DOS v3.2 and later or IBM PC-DOS v4.x can use the APPEND command, as follows:

```
APPEND C:\;C:\DATABASE
```

The APPEND command must be included within the AUTOEXEC.BAT file in a position after the PATH command. A further example of the use of the APPEND command will be given in the next chapter under the **Type Command** section.

Returning to the present discussion, if you are a user of MS-DOS with a version prior to v3.2 or an IBM PC-DOS version prior to v4.x, you cannot use the APPEND command. Instead, you will have to write a special batch file to do the same job. As an example, let us create the DBASE.BAT file, again using the **edlin** line editor, as follows:

```
C:\>edlin \BATCH\DBASE.BAT
New File
*1i
       1:*ECHO OFF
       2:*CLS
       3:*CD\DATABASE
       4:*DB
       5:*CD\
       6:*^C
*e
C:\>_
```

which when activated by typing DBASE, changes the directory to that of DATABASE and executes the DB file. Any other files called from within that file will be found in the correct directory.

The % Replaceable Parameter:

Finally, it would be ideal if the language BASIC could be accessed direct from the root directory. However, we can not include the BASIC directory in the PATH command of the AUTOEXEC.BAT file, as we have done with the DOS and COMMS directories, because there are three main versions of the Basic language, two of which are included in the IBM PC-DOS System disc (BASIC and BASICA; A for advanced), and GWBASIC (which is Olivetti's implementation of the language) which runs on all compatibles.

Instead, we have to create a rather special batch file in the BATCH sub-directory, again using the EDLIN line editor, as shown below:

```
C:\>edlin \BATCH\BAS.BAT
New File
*1i
       1:*ECHO OFF
       2:*CLS
       3:*CD\BASIC
       4:*%1
       5:*CD\
       6:*^C
*e
C:\>_
```

Note the variable %1 in line 4: which can take the name of any of the three Basic languages mentioned above, provided the appropriate variable name is typed after the batch file name. For example, typing

```
C:\>BAS GWBASIC
```

at the prompt, starts executing the commands within the batch file BAS.BAT, but substituting GWBASIC for the %1 variable. Thus, line 4: causes entry into GWBASIC, provided it exists in the BASIC directory. Similarly, typing

```
C:\>BAS BASICA
```

causes entry into BASICA, again provided it exists in the BASIC directory.

Special Batch-file Commands

Apart from all the DOS commands, there are some specific commands which can only be used for batch file processing. These are:

Command	*Action*
CALL	Allows you to call one batch file from within another.
ECHO	Enables or disables the screen display of MS-DOS commands which are being ex-

ecuted from within a batch file, or displays the message that follows ECHO.

FOR

Repeats the specified MS-DOS command for each 'variable' in the specified 'set of items'. The general form of the command is:

```
FOR %%variable IN (set of items) DO command
```

where *command* can include any DOS command or a reference to the %%var. For example,

```
FOR %%X IN (F.OLD F.NEW) DO TYPE %%X
```

will display F.OLD followed by F.NEW.

GOTO label

Transfers control to the line which contains the specified label. For example,

```
GOTO end
---
---
:end
```

sends program control to the :end label

IF

Allows conditional command execution. The general form of the command is:

```
IF [NOT] condition command
```

where 'condition' can be one of

```
EXIST filespec
string1==string2
ERRORLEVEL=n
```

Each of these can be made into a negative condition with the use of the NOT after the IF command.

PAUSE

Suspends execution of the batch file.

REM — Displays comments which follow the REM.

SHIFT — Allows batch files to use more than 10 replaceable parameters in batch file processing. An example of this is as follows:

```
:begin
TYPE %1 | MORE
SHIFT
IF EXIST %1 GOTO begin
REM No more files
```

If we call this batch file SHOW.BAT, then we could look at several different files in succession by simply typing

```
SHOW file1 file2 file3
```

as the SHIFT command causes each to be taken in turn.

Combining Batch Files:

After you have created several batch files, one for each application you load onto your hard disc, plus several others to look at file lists in utility sub-directories you will realize that each such batch file takes up 2 or 4 Kbytes of disc space, depending on the cluster size of your hard disc, even though individual batch files might only be a few bytes in size. To remedy this situation, you could combine all your batch files into one batch file, call it LOAD.BAT, thus saving considerable disc space.

It will be assume here that you have 9 batch files which you would like to combine. These might be BATCH.BAT, DOS.BAT, and NORTON.BAT which produce a listing of the corresponding directories, QA.BAT which loads the Q&A integrated package of word processor and database, QPRO.BAT which loads the Quattro spreadsheet, TURBOC.BAT which loads the Turbo C language, SCALC.BAT, LOTUS23.BAT and LOTUS31.BAT which load the spreadsheets SuperCalc, Lotus 1-2-3 Release 2.3 and Lotus 1-2-3 Release 3.1, respectively.

Before we proceed with the writing of the combined batch file, we shall adapt the SHOW.BAT batch file discussed at the end of the previous section, so that we can obtain a listing on the

printer of the contents of all the batch files we intend to combine into one, thus making our job easier. The new version of the SHOW.BAT batch file, which should be placed in the BATCH sub-directory, is given below.

```
@ECHO OFF
CD\BATCH
:begin
ECHO %1.bat
@ECHO OFF
TYPE %1.bat |MORE >PRN
SHIFT
IF EXIST %1.bat GOTO begin
ECHO No more files
```

Thus, to obtain a listing of the batch files of interest, simply type

SHOW batch dos norton qa qpro turboc scalc lotus23 lotus31

and press <Enter>. Note that the SHOW batch file has been written in such a way as not to require the extension .BAT to be included after the entry of each of its substitution parameters.

Now, with the help of the listing of these batch files, you can proceed to write the combined batch files, which we shall name LOAD.BAT, as follows:

```
@ECHO OFF
IF %1==batch GOTO PL1
IF %1==dos GOTO PL2
IF %1==norton GOTO PL3
IF %1==qa GOTO PL4
IF %1==qpro GOTO PL5
IF %1==turboc GOTO PL6
IF %1==scalc GOTO PL7
IF %1==lotus23 GOTO PL8
IF %1==lotus31 GOTO PL9
GOTO END
:PL1 @echo off
cd\batch
cls
dir/p
GOTO END
:PL2
@echo off
```

```
cd\dos
dir/p
GOTO END
:PL3
@echo off
cd\norton
dir/p
GOTO END
:PL4
@echo off
cd\qa
qa
GOTO END
:PL5
@echo off
cd\qpro
q
GOTO END
:PL6
cd\turboc
tc
GOTO END
:PL7
cd\scalc5
sc5
GOTO END
:PL8
@echo off
cd\123r23
lotus
GOTO END
:PL9
@echo off
cd\123r31
lotus
:END
cd\
```

In the above batch file, we assume that you will be typing the entries corresponding to the substitution parameters in lower case. For example, typing

```
LOAD qpro
```

loads the Quattro Pro package.

If you want to make the batch file respond to both uppercase and lower-case letters, then each line containing the IF statement must be repeated, as shown below:

```
IF %1==qpro GOTO PL1
IF %1==QPRO GOTO PL1
```

and so on.

Adopting the lower-case option only, results in a batch file of 596 bytes, which replaces 9 batch files of 301 bytes of total size. However, by doing so you have saved 16 or 32 Kbytes of disc space, depending on the cluster size of your disc drive, which is a considerable saving.

The COMMAND.COM Processor

This command starts a new command processor that contains all internal commands. This is loaded into memory in two parts: the resident part and the transient part which can be overwritten by some programs in which case the resident part can be used to reload the transient part. The command takes the form:

A>COMMAND [options]

with the following available options:

/E specifies the environment size in bytes, with a default value of 160 bytes

/P prohibits COMMAND.COM from exiting to a higher level

/C executes a following command.

For example, the following statement

```
COMMAND /C CHKDSK A:
```

which might appear in a program starts a new command processor under the current program, runs the CHKDSK command on the disc in the A: drive, and returns to the first command processor.

This command can also be used to allow one batch file to run from within another, although since the arrival of DOS v3.3, this can be done with the use of the CALL command.

6. MANAGING YOUR SYSTEM

MS-DOS provides several commands which help you to manage and control your system's environment. Some of these commands are internal MS-DOS commands and some are external. First we discuss the internal commands.

Changing the Access Date of a File

If your computer is not fitted with a battery backed clock and you have not been entering the correct time and date on booting up the system, then all your saved files will be showing the default date 1/1/80 in the directory entry. To change this date for a given file, set the current TIME and DATE and type

```
C:\>COPY filespec + filespec
```

where filespec stands for drive, path, filename and extension. Ignore the message 'Content of destination file lost before copy' given by MS-DOS when this command has been executed.

The SET Command

To find out what parameters have been set up, type

```
C:\>SET
```

at the prompt which would evoke the response

```
COMSPEC=C:\COMMAND.COM
PATH=C:\;C:DOS;C:\COMMS
PROMPT=$P$G
```

COMSPEC shows which Command Processor is being used by the system, while PATH and PROMPT display the corresponding commands in your AUTOEXEC.BAT file.

The TYPE Command

This command allows you to see on screen the contents of text files. The command takes the form:

```
C:\>TYPE filespec
```

This command is useful because it only lets you have a look at the contents of files without changing the environment in any way.

For example, if you ever wanted to find out what is held in either the CONFIG.SYS or AUTOEXEC.BAT files, then use this command rather than the line editor **edlin**.

If the text file you are looking at is longer than one screen full, then use **Ctrl+S** key sequence (while holding down the key marked **Ctrl**, press the **S** key once) to stop the scrolling of the display. Any key will start the display scrolling again.

Using TYPE on other than ASCII files (such as a .COM or .EXE file) could cause your system to 'hang' as a result of attempting to display certain sequence of machine code that might be contained in the file. If that happens, use the **Ctrl+Alt+Del** key sequence to re-boot the system.

In general, to use the TYPE command to see the contents of files, you must be logged into the sub-directory where the file is found or give the full filespec. For example, to look at the contents of the DOS.BAT file (which is to be found in the BATCH sub-directory) when at the C:\> prompt, you will have to use:

```
C:\>TYPE \BATCH\DOS.BAT
```

Alternatively, if you include the BATCH sub-directory in an APPEND command following the PATH command within the AUTOEXEC.BAT file, then you could omit the sub-directory in the TYPE command. Thus, the BATCH sub-directory must be included in the PATH command to be able to find and execute the various batch files, but in order to find any of these files from within a DOS command, such as TYPE, the sub-directory must also be included in the APPEND command.

The TYPE command could be used to direct text files to the printer by typing

```
C:\>TYPE EXAMPLE.TXT >PRN
```

where PRN stands for 'printer' which is connected to the parallel printer port.

The VER & VOL Commands

To find out which version of MS-DOS/PC-DOS you are currently using, type

```
C:\>VER
```

at the prompt.

To find out the volume label of the disc in the logged drive, type

```
C:\>VOL
```

at the prompt. If the disc was not labelled during formatting, then the computer will respond with

```
Volume in drive B has no label
```

otherwise the appropriate label will be displayed.

The MORE & SORT Filter Commands

The MORE external filter command allows you to view text files a page (23 lines) at a time - you are prompted to press a key to display the next page. As such, it can be combined with other commands to control scrolling of long ASCII files. For example,

```
C:\>TYPE EXAMPLE.TXT |MORE
```

or even used by itself (giving quicker response) as

```
C:\>MORE<EXAMPLE.TXT
```

can help you with viewing long text files if you are not used to or quick enough to use the **Ctrl+S** key sequence to halt scrolling, after issuing the TYPE command.

One of the ways in which the SORT external command can be used is to sort and display alphabetically the contents of a directory. For example,

```
C:\>DIR |SORT
```

will sort the contents of the logged directory, including the header and footer information, and display the result. For long directories, use this command together with the MORE filter, as follows:

```
C:\>DIR |SORT |MORE
```

to display the sorted directory a page at a time. A hard copy of the sorted directory of a disc could be obtained by typing

```
C:\>DIR |SORT >PRN
```

which re-directs output through the parallel printer port.

The PRINT & PRTSC Commands

The first time the PRINT command is used it has to be loaded into memory as it is an external MS-DOS command. However, from then on it resides in memory and can be used without having to re-load it.

The PRINT command provides background printing, that is, it can print long files while you are doing something else with your computer. In fact, using this command provides you with a print spooler which allows you to make and control a queue of several files for printing. The command takes the form:

`C:\>PRINT filespec`	adds filespec to print queue
`C:\>PRINT filespec /C`	cancels printing that file
`C:\>PRINT /T`	terminates all printing
`C:\>PRINT`	displays files in queue

The PRINT command assumes that you have continuous paper in your printer. There is no facility to pause printing.

To print the two text files TEXT1.DOC and TEXT2.DOC, type

```
C:\>PRINT TEXT1.DOC
C:\>PRINT TEXT2.DOC
```

Wildcard characters can also be used in the command, as follows:

```
C:\>PRINT TEXT*.DOC
```

which will spool all the files starting with the characters TEXT and having the extension .DOC to the printer.

Text which is displayed on the screen can be sent to the printer by pressing the Print Screen (**Shift+PrtSc**) key.

On the other hand, pressing the **Ctrl** and **Print_Screen** keys simultaneously causes re-direction of output to the printer. To cancel the effect, repeat the same key stroke.

The BACKUP & RESTORE Commands

The external BACKUP command allows you to archive files from the hard disc. Since your disc contains valuable work, you must make additional copies of all your important files. The BACKUP utility allows you to generate those back-up copies on floppy discs. If you have a hard disc, you should use this utility often; daily if necessary. The command takes the form:

BACKUP *source destination options*

where *source* is the drive/path/files to be backed up,
destination is the drive to back-up to, and
options are:

/A to add the files to a disc in the destination drive
/D:*date* to back-up only files from the specified date onwards

/M to back-up only files modified since they were last backed up

/S to also back-up sub-directories of the source path.

Thus, to back-up, for the first time, all the word processor files whose path is \DATABASE\WPFILES, we type

```
C:\>BACKUP C:\DATABASE\WPFILES\*.* A:
```

while to back-up only files modified since they were last backed up, we type

```
C:\>BACKUP C:\DATABASE\WPFILES\*.* A:/M
```

In both cases, the wildcard characters *.* ensures that all files with all their extensions in the WPFILES sub-directory are backed up.

The RESTORE external command allows you to de-archive files. It is the only utility which can restore to the hard disc files previously copied to floppy discs using the BACKUP utility. The command takes the form:

RESTORE *source destination options*

where *source* is the drive to restore from,
destination is the drive/path/files to restore, and
options are:

/P to prompt Y/N? before restoring, and
/S to also restore files from sub-directories.

Thus, typing

```
C:\>RESTORE A: C:\DATABASE\WPFILES\*.*/P
```

restores selected files from the floppy disc in the A: drive to the sub-directory WPFILES in the C: drive, provided these files were backed up from the same named sub-directories.

***

MS-DOS has many more commands which can be used to control a micro in special ways. However, this is an area which lies outside the scope of this book. What was covered here, together with the summary of the DOS commands given in the penultimate section of this book, is more than enough to allow effective control of a microcomputer.

If you would like to be able to write customised batch files, create specialist programs with the use of the **debug** program and learn how to design your own professional looking menu screens, then may I suggest that you refer to another book, entitled *A Concise Advanced User's Guide to MS-DOS (BP 264)*, also published by Bernard Babani (publishing) Ltd.

***

7. MS-DOS VERSION 4.X

PC/MS-DOS version 4.x ends the limit of 32MB partition on hard discs by employing a 32-bit File Allocation Table (FAT), instead of the previous 16-bit employed in earlier versions, with the limit of single partitions now being stretched to 4GB. If one of these partitions is less than or equal to 32MB, then it would be possible to boot up the system from that drive with PC/MS-DOS v3.x, otherwise not. The SHARE program is automatically installed if a hard drive with greater than 32MB is set, so programs that will not work with SHARE installed will not run under PC/MS-DOS v4.x on a large hard disc.

If you are running version 4.x and, on booting the system, you get a warning message about SHARE not being installed while using large media, then edit your CONFIG.SYS file and include the line

```
INSTALL=C:\DOS\SHARE.EXE
```

but providing the correct path where SHARE.EXE can be found on your hard disc, if different from the above.

MS-DOS v4.x supports the expanded memory specification (EMS), as developed by Lotus, Intel and Microsoft, which breaks the 640kB barrier by exploiting the expanded memory boards to store data associated with the BACKUP, FASTOPEN and VDISK commands. These are made possible with the use of two expanded memory drivers - XMA2EMS.SYS and XMAEM.SYS. The first is for use with systems based on the 8086, 8088 and 80286 processors, while the second is for use with the 80386 processor.

Other improvements include extra video mode settings to support displays of greater than 25 lines, as provided by enhanced (EGA) and video (VGA) cards, and graphic screen printing for EGA and VGA graphics adaptors. The GRAPHICS program now allows the <PrtSc> key to perform a graphics dump from EGA and VGA screens to a variety of IBM printers. The commands required to perform these can be modified for other non-IBM printers. You specify your printer during installation, but make sure that the printer is switched on before booting up the system with DOS v4.x, otherwise you'll be told by the installation program that the printer of your choice is not supported!

Further additions or modifications to PC/MS-DOS commands are described in the section 'Command Summary' with v4.x commands being marked with a superscripted asterisk (*). Perhaps it is worth mentioning that the resident parts of v4.x require an additional 15kB of memory above and beyond the requirement of v3.3, with an additional 300kB of memory for loading the DOS shell which however is overwritten when programs are loaded. The use of the DOS shell, which is optional, will be discussed shortly.

Finally, error messages have been enhanced so as to give some indication as to what might be wrong, which is a vast improvement on previous versions of the operating system.

Installing MS-DOS 4.x

To install MS-DOS v4.x, you can use the SYS command on discs with a previous installation of DOS, but newcomers or those who want to re-define a drive with more that 32MB will have to format their disc. A new menu-driven program, called SELECT, guides the user through installation and automatically creates the necessary AUTOEXEC.BAT and CONFIG.SYS files. To use this program, insert the 'Install' disc in the A: drive and use the key combination **Ctrl+Alt+Del** to re-boot the system from the A: drive.

After the introductory information screens in which it is explained that in order to install DOS v4.x one requires a blank floppy disc, DOS asks the user to choose between three levels of installation: minimum, balanced and maximum DOS functionality, all of which are dependent on the amount of memory available. Choosing any of these only varies the amount of information that the installation program writes to the AUTOEXEC.BAT and CONFIG.SYS files, with the maximum DOS functionality choice requiring about 7kB more memory. However, since the files thus created are unlikely to fit the needs of users with already implemented systems, the choice of DOS functionality becomes irrelevant. Such users can continue to use their old AUTOEXEC.BAT and CONFIG.SYS files, but with the necessary additions to run the DOS shell, if they choose to do so.

SELECT then asks the user about the type of keyboard, printer and the drive and directory onto which to install DOS v4.x, at which point various configuration choices are offered

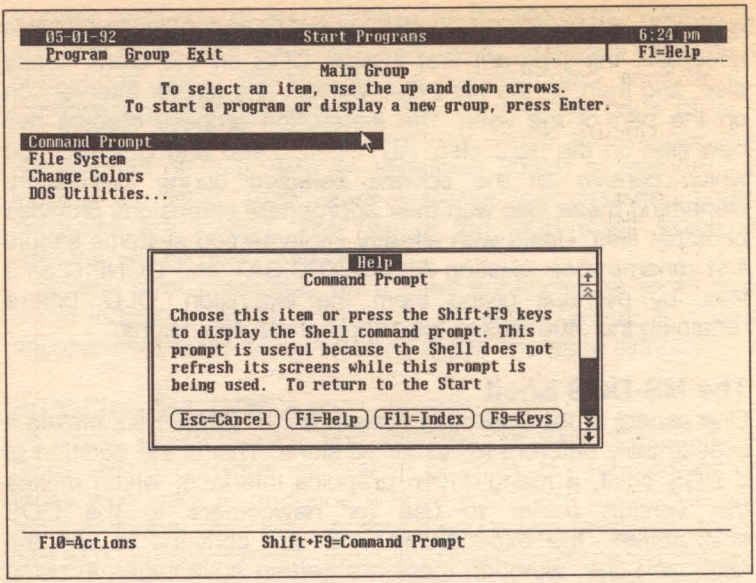
regarding the AUTOEXEC.BAT and CONFIG.SYS files. Once accepted, the program first installs DOS on the blank floppy disc, and then on the hard disc without any further intervention on the part of the user. The installation process creates two new files on the hard disc: AUTOEXEC.400 and CONFIG.400 which contain all the options selected during installation. Renaming these files with their appropriate extensions provides bootable files. Users with already implemented systems should first rename their existing AUTOEXEC.BAT and CONFIG.SYS files, by perhaps giving them the extension .OLD, before renaming the .400 files to their appropriate extensions.

The MS-DOS Shell

One aspect of the operating system under version 4.x makes it substantially different to earlier versions. This is the addition of a DOS shell, a menu-driven graphics interface, which makes this version easier to use for newcomers to the DOS environment; the experienced user can continue to use the command line, although there are certain advantages in using the DOS shell, such as the ability to rename directories.

The menus employed in the DOS shell are consistent with those used in Microsoft Windows (prior to version 3.0) which will make it easier for those who become familiar with it to migrate to the OS/2's Presentation Manager - a more advanced operating system which seeks to replace DOS. However, as OS/2 will only run on PCs based around the 80286 and 80386 processors, the twelve million users with machines based on the 8086 and 8088 processors will be confined to the PC/MS-DOS operating system. Thus, the DOS shell of v4.x was designed to provide the closest link, be it only an artificial one, between DOS and OS/2.

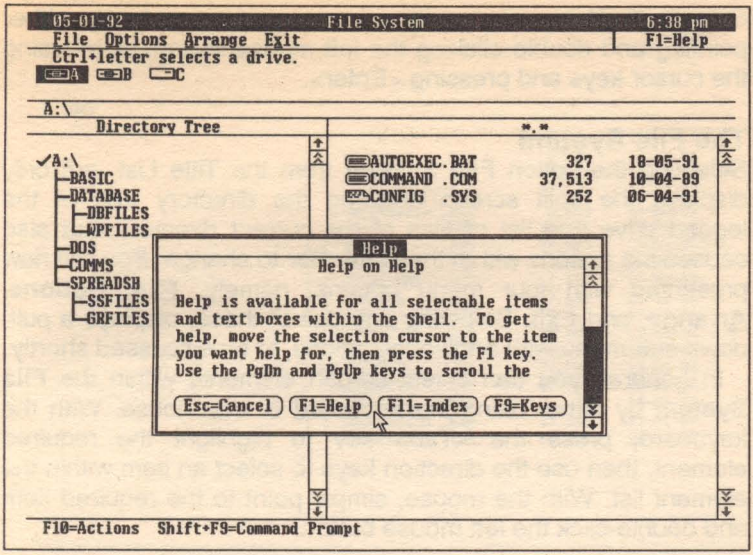
When the program first loads, normally as a result of the last entry in the AUTOEXEC.BAT in the name of DOSSHELL, the user is presented with the Start Program screen, as shown on the next page. The first line of the display is the Title Bar, which shows the date, title of the program being used, and time. Below this is the Menu Bar with the captions **Program**, **Group** and **Exit**. This menu bar can be accessed with the use of the **F10** function key, with **F1** providing context sensitive help. Selection of any one of these three options causes the display of a pull-down sub-menu.


05-01-92
Start Programs
6:24 pm
Program Group Exit
F1=Help
Main Group
To select an item, use the up and down arrows.
To start a program or display a new group, press Enter.
Command Prompt
File System
Change Colors
DOS Utilities...
Help
Command Prompt
Choose this item or press the Shift+F9 keys to display the Shell command prompt. This prompt is useful because the Shell does not refresh its screens while this prompt is being used. To return to the Start
Esc=Cancel
F1=Help
F11=Index
F9=Keys
F10=Actions
Shift+F9=Command Prompt

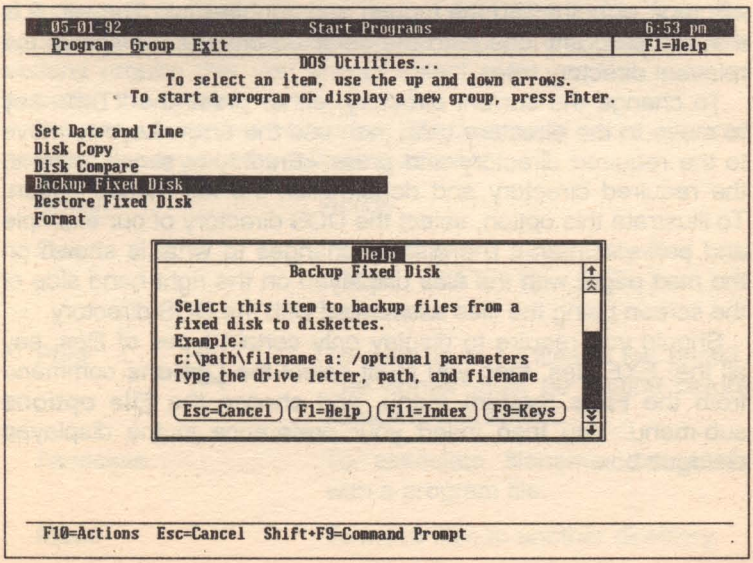
The first two of these options allow the user to add applications to the program list, change their information, or create more than one list. However, to achieve these, the user must be capable of writing short code reminiscent of that of batch files.

On the main part of the screen is a menu of options, called the Title List, the first of which is **Command Prompt**. Choosing this option causes the Start Program to shell out to the DOS command line. To return to the DOSSHELL, you must type **Exit** at the DOS prompt.

The second menu entry is the **File System** which is similar to the one supplied with Presentation Manager. It displays the user's directories and sub-directories on the screen, together with a list of appropriate files. Selecting this option causes a divided screen to be displayed, as shown on the next page. On the left of the screen is a directory tree of the logged drive, while on the right is a list of files within the current directory. Each of the files has an icon alongside it that indicated whether it is an executable file. The option also allows the user to add directories, sub-directories and files, copy, delete, or rename them.



The third option allows the user to change screen colours, while the fourth allows access to the DOS Utilities such as XCOPY, BACKUP & RESTORE, and FORMAT, as shown below:



Selection of one of these options can be made by either pointing and double clicking the left mouse button, or by using the cursor keys and pressing <Enter>.

The File System

Selecting the option **File System** from the Title List, not only displays the split screen showing the directory tree of the logged drive and list of files of the current directory, but also causes the options within the Menu Bar to change. You are now presented with four menu options, namely, **File**, **Options**, **Arrange**, and **Exit**. Selecting any one of these, displays a pull-down sub-menu with additional options, to be discussed shortly.

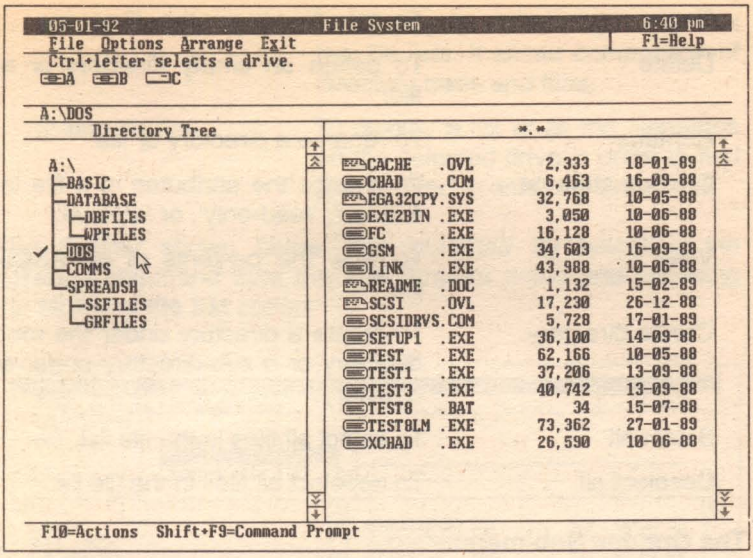
In general, you can select screen elements within the **File System** by either using the keyboard or the mouse. With the keyboard, press the <Tab> key to highlight the required element, then use the direction keys to select an item within the element list. With the mouse, simply point to the required item and double-click the left mouse button.

Changing the Directory & Logged Drive:

To change the logged drive, either press the <Tab> key which causes the selector pointer to move to the drive identifiers, then use the arrow keys to select a different drive and press <Enter>, or point with the mouse and double-click. The result is that the program logs onto the selected drive and displays the relevant directory tree.

To change the current directory, either press the <Tab> key to move to the directory tree, then use the arrow keys to move to the required directory and press <Enter>, or simply point to the required directory and double-click the left mouse button. To illustrate this option, select the DOS directory of our example and press <Enter>. The display changes to what is shown on the next page, with the files displayed on the right-hand side of the screen being the files associated with the DOS directory.

Should you require to display only certain types of files, say all the .EXE files, then you must select the **Options** command from the **Files System** menu, and choose the **File options** sub-menu. You then insert your preference in the displayed dialogue box.



The File Sub-menu:

Highlighting a file (such as the AUTOEXEC.BAT) and then selecting the **File** command, causes DOSSHELL to display a pull-down sub-menu. If you do not highlight a file first, the options in the pull-down sub-menu appear dim. Most of the options require that you should select a file, or files, prior to selecting the option.

The various options within the **File** command sub-menu can be used to carry out the following actions:

Option	*Action*
Open	To open (start) a file. Only files with .COM, .BAT, or .EXE can be started.
Print	To print up to 10 files to the printer. The printer must be on-line before selecting this option.
Associate	To associate filename extensions with a program file.
Move	To move files to another directory.

Copy	To copy files to another directory.
Delete	To delete an empty directory or a file.
Rename	To rename a directory or file.
Change attribute	To change the attributes of a file to 'hidden', 'read-only', or 'archive'.
View	To view the contents of an ASCII file.
Create directory	To create a directory under the root directory or a sub-directory under a directory.
Select all	To select all files in the file list.
Deselect all	To deselect all files in the file list.

The Options Sub-menu:
The various options within the **Options** command sub-menu can be used to carry out the following actions:

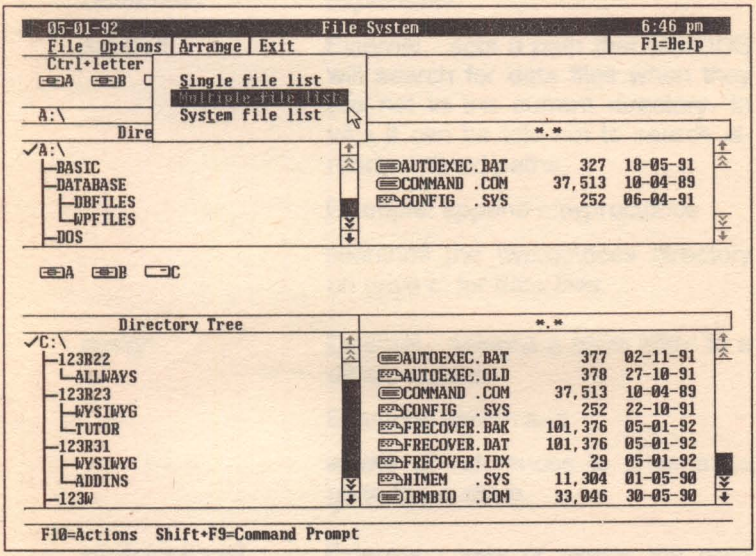
Option	*Action*
Display options	To specify a filename or to sort the file list by name, date, size, extension, or disc order.
File options	To request or cancel file confirmation prompts and global file selection.
Show information	to show the status information of the selected drive, directory, and file.

The Arrange Sub-menu:
The various options within the **Arrange** command sub-menu can be used to carry our the following actions:

Option	*Action*
Single file list	To display the directory tree and file list for the currently selected drive.

Multiple file list	To display two directory trees and two file lists. It allows comparison of directory trees and files.
System file list	To display a list of all the filenames on the selected drive in order to find files located in different directories.

The display shown below was obtained by selecting the **Arrange** command from the **File System** menu and choosing the **Multiple file list** option.



No doubt new users will find DOSSHELL very useful, while experienced users will find the ability to rename directories extremely convenient.

8. COMMAND SUMMARY

The following is a summary of the commands supported by the MS-DOS operating environment. For a fuller explanation of both commands and options, consult your system's PC/MS-DOS reference manual. The various commands are labelled internal or external, with external commands being accessible to the user only if the full filespec (drive and path) is given to where the appropriate command file resides. Improvements or additions in v4.x are marked with a superscripted asterisk (*).

Command	*Explanation*
append*	External - sets a path that MS-DOS will search for data files when they are not in the current directory. In v4.x it can be told not to search already defined paths. Example: append c:\wproc\docs searches the \wproc\docs directory on drive c: for data files.
assign	External - assigns a drive letter to a different drive. Example: assign a=c allows all references to drive a: to go to the c: drive.
attrib [options]	External - sets or resets the *read only* attribute & archive bit of a file, and displays the attributes of a file. Options: +r sets read-only mode of a file -r disables read-only mode +a sets the archive bit of a file -a clears the archive bit. Example: attrib +R filespec

backup [options]*

External - backs up one or more files from one disc to another. In v4.x it can also automatically format the destination disc.

Options:

/s backs up sub-directory files to file in current directory

/m includes files that have been changed since last backup

/a adds files to be backed up to those already on the backup disc without erasing old files

/d backs up only those files which were modified after that date

/l makes a backup log entry in a file called BACKUP.LOG.

Example: backup c:\ a:/s

backs up all files on the c: drive onto the a: drive.

break

Internal - sets the Ctrl+Break or the Ctrl+C option.

Example: break ON

cd (or chdir)

Internal - changes the working directory to a different directory.

Example: cd\wproc\docs

chcp [nnn]

Internal - selects current code page for as many devices as possible. Omitting *nnn* displays the current code page.

chkdsk [options]*

External - analyses the directories, files, and File Allocation Table on the logged or designated drive and produces a status report. In v4.x it also reports the volume, serial number and disc allocation units.

Options:

/f fixes any problems found during the check

/v causes the display of filespecs as they are being processed.

Example: chkdsk a:/f/v

cls — Internal - clears the screen.

command [options] — External - starts the command processor which is loaded into memory in two parts; the resident part and the transient part. If the transient part is overwritten by a program, it is reloaded.

Options:

/e specifies the environment size in bytes (default = 160 bytes)

/p prohibits command.com from exiting to a higher level

/c executes a following command

Example: command /c chkdsk a:

starts a new command processor under the current program, runs the chkdsk command on the disc in the A: drive, and returns to the first command processor.

comp — External - compares two files and reports any differences.

Example: comp file1 file2

copy [option] — Internal - copies one or more files to specified disc. If preferred, copies can be given different names.

Option:

/v causes the verification of data written on the destination disc.

Example: copy *.exe a:/v

copies all files with the .exe extension to the a: drive with verification.

ctty — Internal - changes the standard I/O console to an auxiliary (aux) console, and vice versa.

Example: ctty aux

moves all input/output from the current device (console) to an aux port such as another terminal. The command *ctty con* moves I/O back to the console.

date — Internal - enters or changes the current date.

del [option]* — Internal - deletes all files with the designated filespec.

Option v4.x only:
/p displays filenames to confirm deletion.

Example: del a:*.txt

deletes all files which have the extension .txt from the a: drive.

dir — Internal - lists the files in a directory.

diskcomp — External - compares the contents of the disc in the source drive to the disc in the destination drive.

diskcopy — External - copies the contents of the disc in the source drive to the disc in the destination drive.

dosshell* — External - a new v4.x command which activates the front-end graphical interface.

exe2bin — External - converts .exe files to binary format.

exit — Internal - exits the command processor and returns to a previous level.

fastopen [option]* — External - store in memory the location of directories and recently opened files on a specified drive.

Option:

/x allows use of expanded memory. If this option is used, then the **/x** option must also be used with the **buffers** command.

fdisk* — External - sets up and partitions the fixed disc for use with MS-DOS and other operating systems. This command is also used to display and change the current active partition. Version 4.x supports an 80-column screen. It also has improved user-friendly commands to allow disc partitioning in megabytes or percentages instead of cylinders.

fdiskoff — External - could have another name, but it use is to park the fixed disc heads. This should be done before moving a computer equipped with a hard disc to prevent disc damage.

find [options] — External - searches for a specific string of text in a specified ASCII file or files.

Options:

/v displays all lines not containing the specified string

/c prints the count of lines containing the string

/n precedes each occurrence with the relative line number in the file.

Example: find "lost words" chap1

searches for the string *lost words* (which must appear within full quotes) in the named file (chap1).

format [options]* External - formats the disc in the specified drive.

Options:

/8 formats with 8 sectors per track

/4 formats a double-sided disc with 40 tracks, 9 sectors per track for 360 KB in a high capacity (1.2 MB) disc drive per track

/n specifies the number of sectors per track, i.e. /n:9 for nine sectors

/t specifies the number of tracks, i.e. /t:40 for forty tracks

/v allows a volume label to be given to the disc

/s copies the system files from the logged drive.

Version 4.x also uses:

/f:size to specify disc capacity

/v:label to allow specification of *label* without prompting after formatting.

Example: format a:/4/s

graftabl*	External - loads a custom designed, colour graphics font table into memory. In v4.x, the multilingual code page 850 is now supported.
graphics*	External - v4.x now supports EGA and VGA graphics modes to provide screen dumps to IBM Graphics, Proprinters and compatibles.
install*	External - a new v4.x command which provides an improved method of loading memory-resident pop-up programs in addition to certain DOS commands.
join	External - joins a disc drive to a specific path.
keyb [xx]	External - selects a special keyboard layout. Omitting **xx** returns the current status of the keyboard.
label	External - creates or changes the volume identification label on a disc.
md (or mkdir)	Internal - creates a new directory on the specified disc.
mem*	External - a new v4.x command which reports the amounts of conventional, expanded and extended memory that are available.
mode [options]*	External - sets the mode of operation on a display monitor, parallel/serial printer or the RS232C port. In v4.x the keyboard repetition and auto-repeat start delay time can be set. Also, it allows the setting of the number of rows to any of 25, 43 or

50 on the screen, and there is a wider range of serial-port configurations.

Options:

Display: mode [n]

40	sets display width to 40 characters per line
80	sets display width to 80 characters per line
bw40	sets screen to black and white display with 40 characters
bw80	sets screen to black and white display with 80 characters
co40	sets screen to colour display with 40 characters
co80	sets screen to colour display with 80 characters
mono	sets screen to monochrome with 80 characters.

Printer: mode LPTi: [n][,[m][,p]]

i	sets printer number with legal values from 1 to 3
n	sets number of characters per line with legal values of 80 or 132
m	sets the number of lines per inch with legal values of 6 or 8
p	allows continuous re-entry on a time-out error.

Example: mode LPT1: 132,8

sets the printer in the first parallel port to 132 characters per line and 8 lines per inch.

Serial printer: mode LPTi: = COMj
This command redirects all output sent to one of the parallel printer ports to one of the serial (RS232C) ports. Before using this command, the serial port must be initialized using the *p* option of the printer mode command.

i	sets printer number with legal values from 1 to 3
j	sets the serial port with legal values of 1 or 2.

more

External - reads data from the standard I/O and sends output to the console one screen-full at a time.

Example: type read.me | more

displays the contents of the read.me file one screen at a time.

nlsfunc

External - provides support for extended country information and allows the use of **chcp** command to select code pages for all devices defined as having code page switching support.

now

External - displays the current date and time.

path

Internal - sets and displays the path to be searched by DOS for external commands or batch files.

Example: path c:\;c:\dos;c:\comms

will search the root directory as well as the dos and comms sub-directories for files with .COM, .EXE, and .BAT extensions.

print [options]

External - can be used to print text files in background mode, while other tasks are being performed. Using the command without options displays files already in the print queue.

Options:

/d specifies the print device such as PRN or AUX

/b sets size of internal buffer with legal values from 512 to 16384 bytes, speeding up printing

/q specifies the number of files in the print queue, normally 10, with legal values from 4 to 32

/t allows cancellation of files in print queue.

prompt

Internal - changes the command prompt.

Example: pg

which allows the path of the current working directory to be displayed as the prompt.

rd (or rmdir)

Internal - removes the specified directory.

recover

External - recovers a file or an entire disc containing bad sectors.

ren (or rename)

Internal - changes the file name.

Example: ren a:\docs\mem1 mem2

will rename the mem1 file, which is to be found in sub-directory docs on a disc in the a: drive, to mem2.

replace [options]*

External - allows easy updating of files from a source disc to a target disc of files having the same name.

Options:

/a also adds files that exist on the source disc but not on the target disc

/d replaces files in the destination disc only if the source files are newer

/p prompts the user before replacing

/r replaces read only files.

Version 4.x also uses:

/u updates files with a time and date on the source disc more recent than those on the destination disc.

restore [options]

External - restores one or more files that were backed up using the *backup* command.

Options:

/s restores files in the specified directory and all files in any sub-directories of the specified directory

/p prompts user before overwriting an existing file by restoring.

select*

External - sets the formats for date and time, currency sign and decimal point convention. In v4.x there are full-screen installation and help routines.

set

Internal - sets strings into the command processor's environment. The general form of the command is:

	set [name=[parameter]] Set by itself displays the current environment.
share	External - installs file sharing and locking.
sort [options]	External - reads data from the console or a file, sorts it and sends it to the console or file. Options: /r sorts in reverse order. Example: dir | sort sorts the output of the *dir* command in alphabetical order.
subst	External - allows substitution of a virtual drive for an existing drive and path. Example: subst d: a:\wproc\docs will cause future reference to drive d: to be taken as replacement to the longer reference to a:\wproc\docs.
switches*	External - a new v4.x command which forces the conventional keyboard layout on to an enhanced keyboard.
sys*	External - transfers the PC/MS-DOS system files from the logged drive to the disc in the specified drive. Version 4.x allows the specification of source drive and path commands to transfer system files across a network.

time* — Internal - displays and sets the system time. In v4.x a 12- or 24-hour format is supported.

tree* — External - displays the directory structure. In v4.x the display is shown graphically.

type — Internal - displays the contents of a file on the console.

vdisk — External - allows the setting up of RAM memory as a virtual disc. In v4.x the /x option can be used to allow the virtual disc to reside in expanded memory.

ver — Internal - displays the PC/MS-DOS version number.

verify — Internal - allows the verify switch to be turned ON or OFF.

Example: verify OFF

vol — Internal - displays the disc volume label, if it exists.

xcopy [options] — External - copies files and directories, including lower level sub-directories, if they exist, to the destination drive and directory.

Options:

/a copies source files that have their archive bit set

/d copies source files which were modified on or after a specified date

/e copies sub-directories even if they are empty - use this option in conjunction with /s

/m copies archived files only, but also turns off the archive bit in the source file

/p prompts the user with '(Y/N?)'

/s copies directories and their sub-directories unless they are empty

/v causes verification of each file as it is written.

9. GLOSSARY OF TERMS

ASCII	It is a binary code representation of a character set. The name stands for 'American Standard Code for Information Interchange'.
AUTOEXEC.BAT	A batch file containing commands which are automatically executed on booting up the system.
BACKUP	To make a back-up copy of a file or a disc for safekeeping.
BASIC	A high level programming language. The name stands for 'Beginner's All-purpose Symbolic Instruction Code'.
BASICA	IBM's implementation of BASIC. The A stands for 'advanced'.
Batch file	An ASCII formated file that contains MS-DOS commands which can be executed by the computer.
BIOS	The Basic Input/Output System. It allows the core of the operating system to communicate with the hardware.
Boot	To start up the computer and load the DOS operating system.
Booting up	The process of starting up the computer.
Buffer	RAM memory allocated to store data being read from disc.
Byte	A grouping of binary digits (0 or 1) which represent information.
Cluster	A unit of one or more sectors. It is the minimum amount of space that can be allocated to a file on disc.
Click	To quickly press and release a mouse button.

Cold boot	The process of starting your PC by turning on the power switch.
COMMAND.COM	The Operating System's Command Processor which analyzes what is typed at the keyboard and causes execution of appropriate commands.
Command line	The line on the computer's screen into which you enter DOS commands.
Command Prompt	The prompt (C>) which appears on the computer's screen to let you know that MS-DOS is ready to receive a command.
CONFIG.SYS	A special file that allows the system to be configured closer to requirement.
CPU	The Central Processing Unit; the main chip that executes all instructions entered into a computer.
cursor	The blinking line indicating where the next input can be entered.
Default	The command, device or option automatically chosen by the system.
device driver	A special file that must be loaded into memory for DOS to be able to address a specific procedure or hardware device. These are normally installed from the CONFIG.SYS file at system start-up.
device name	A logical name used by DOS to identify a device, such as LPT1 or COM1 for the parallel or serial printer.
Dialogue box	A box that MS-DOS displays on the screen when DOSSHELL is in operation, to ask the user for more information.
Directory	An area on disc where information relating to a group of files is kept.

Directory identifier	Displays the active disc drive and directory on the File System screen of DOSSHELL.
Directory tree	A pictorial representation of your disc's structure.
Disc file	A collection of program code, or data, that is stored under a given name on a disc.
DOS	The Disc Operating System. A collection of small specialised programs that allow interaction between user and computer.
DR-DOS	Digital Research's implementation of the Disc Operating System for compatible PCs.
DOSSHELL	The name of the front-end graphical interface provided by MS-DOS v4.x.
Double-click	To quickly press and release a mouse button twice.
Drag	To press and hold down the left mouse button while moving the mouse.
Drive name	The letter followed by a colon which identifies a floppy or hard disc drive.
Edlin	The MS-DOS line editor used to create and modify ASCII formatted files, such as batch files and the CONFIG.SYS file.
Enter key	The key that must be pressed after entering data.
Expanded memory	This is memory outside the conventional RAM (0-640Kbytes) that DOS uses. It can be used by software to store data and run applications.
Extended memory	This is memory above the 1-megabyte memory address which DOS can use for limited operations.

External command	A command DOS executes by first loading it from an external disc file.
FAT	The File Allocation Table. An area on disc where information is kept on which space on disc has been allocated to which file.
File	The name given to an area on disc containing a program or data.
File list	A list of filenames contained in the active directory.
Filename	The name given to a file. It must not exceed 8 characters in length and can have an extension of up to 3 characters.
Filespec	File specification made up of drive, path, filename and a three letter extension.
Fixed disc	The hard disc.
Floppy disc	A flexible removable disc on which information can be stored magnetically.
Formatting	The process of preparing a disc so that it can store information. During formatting, sectors, tracks, a directory, and the FAT are created on the disc.
FORTRAN	A high level programming language. It stands for FORmula TRANslation.
Function key	One of the series of 10 or 12 keys marked with the letter F and a numeral, used for specific operations.
GUI	A Graphical User Interface, such as that of DOSSHELL which uses visual displays to eliminate the need for typing commands.
GW-BASIC	Olivetti's implementation of BASICA programming language for compatible PCs.
Hard disc	A device built into the computer for holding programs and data. It is sometimes referred to as the fixed disc.

Hardware	The equipment that makes up a computer system, excluding the programs or software.
Hidden files	Files that do not normally appear in a directory listing, such as the IO.SYS and MSDOS.SYS files.
Internal command	One of a set of many command available to you at any time as they are loaded into memory every time you start your PC.
Key combination	When two or more keys are pressed simultaneously, such as Ctrl+Alt+Del.
Kilobyte	1024 bytes.
LABEL	The MS-DOS command which allows you to give a name to your disc.
Main Group	The list of options displayed on the Start Programs screen.
Megabyte	1024 kilobytes.
Memory	Part of computer consisting of storage elements organised into addressable locations that can hold data and instructions.
Menu	A list of available options as appears in the DOSSHELL.
Menu bar	The horizontal bar that lists the names of menus.
Monitor	The display device connected to your PC.
Mouse	A device used to manipulate a pointer around your display and activate a certain process by pressing a button.
MS-DOS	Microsoft's implementation of the Disc Operating System for compatible PCs.
Operating System	A group of programs that translates your commands to the computer.

Parameter	Additional information appended to an MS-DOS command to indicate how the command should be executed.
PATH	The drive and directories that DOS should look in for files.
PC-DOS	IBM's implementation of the Disc Operating System for IBM PCs.
Peripheral	Any device attached to a PC.
Port	An input/output address through which your PC interacts with external devices.
Program	A set of instructions which cause the computer to perform certain tasks.
Prompt	The System prompt displayed on screen, such as A> or C>, indicating that DOS is ready to accept commands.
RAM	Random Access Memory. The micro's volatile memory. Data held in it is lost when power is switched off.
Return key	The same as Enter key.
ROM	Read Only Memory. The micro's non-volatile memory. Data are written into this memory at manufacture and are not affected by power loss.
Root directory	The main disc directory under which a number of sub-directories can be created.
Scroll bar	The bar that appears at the right side of the File List and Directory sections of the File System screen of DOSSHELL.
Sector	Disc space, normally 512 bytes long.
SHELL	A front end to MS-DOS or an alternative Command Processor.
Software	The programs and instructions that control your PC's functionality.

Start Program	The program that loads first when you load DOSSHELL. It contains the Main Group programs.
SYSTEM	A BASIC command which causes exit from BASIC and return to MS-DOS.
System	Short for computer system, implying a specific collection of hardware and software.
System disc	A disc containing MS-DOS' three main files and other Utilities.
System prompt	The prompt displayed on the screen, such as A> or C> indicating that DOS is ready to accept commands.
Title bar	A horizontal bar across the top of each screen in DOSSHELL that contains the name of the screen.
Title List	A list of programs found on the Main Group of the Start Programs screen of DOSSHELL.
Volume label	An identifying label written to a disc when a disc is first formatted.
Warm boot	The process of starting your PC with the use of the Ctrl+Alt+Del key combination.
Wildcard character	A character that can be included in a filename to indicate any character (?) or group of characters (*) that might match that position in other filenames.

INDEX

ANSI.SYS 40
APPEND command .. 45, 69
ASCII files 31
AUTOEXEC.BAT file 5, 23, 42
ASSIGN command 69
ATTRIB command 69

BACKUP command .. 56, 70
BASIC 10, 22, 47
Batch
  file commands 47
    CALL 47
    ECHO 47
    FOR 48
    GOTO 48
    IF 48
    PAUSE 48
    REM 49
    SHIFT 49
  files 44
    use of KEYB UK 44
    use of CLS 44
    use of ECHO 44
    use of PATH 44
    use of PROMPT 44
    use of VER 44
    use of % parameter 46
BIOS 3
Booting up 3
BREAK command . 6, 38, 70
BUFFERS 37, 38
Built-in commands 5

CD command 6, 23, 70
CHCP command 70
CHKDSK command .. 19, 70
CLS command 6, 71
CODEPAGE command ... 39
COM.SYS 40
COMMAND.COM .. 3, 52, 71
Command processor 3
Commands, built-in 5
COMP command 17, 71
Comparing discs 17
Comparing files 17
CONFIG.SYS 5, 37
Configuration commands . 38
Configure the system 37
COPY command .. 6, 16, 71
Copy mode (EDLIN) 34
COUNTRY 39
Ctrl+Alt+Del 5, 38
Ctrl+C 32
Ctrl+S 54
Ctrl+Z 35
Ctty command 72
Current line (EDLIN) 32

DATE command ... 6, 53, 72
DELete command . 6, 18, 72
Delete mode (EDLIN) 34
DEVICE 40
DIR 7, 72
  /P option 9
  /W option 9
  wildcard characters 10
Directing I/O 54
Directories 21
Directory
  listing 25
  tree 26
Disc capacity 14
DISKCOMP command 17, 72
DISKCOPY command 16, 72
DOS
  SHELL command . 61, 73
  Versions 1
  v4.0 59
DRIVER.SYS 40

ECHO command 6, 47
Edit mode (EDLIN) 33
EDLIN 31
EGA.SYS 40
EXE2BIN command 73
EXIT command 73
Exit EDLIN 36
External commands 7

FASTOPEN command ... 73
FAT 19
FCBS command 41
FDISK command 4, 73
FDISKOFF command 73
FILES 37, 44
Filenames 3, 8
Filters 55
FIND command 73
FOR command 48
FORMAT command .. 13, 74

GOTO command 48
GRAFTABL command 75
GRAPHICS command 75

IBMBIO.COM 3
IBMDOS.COM 3
IF command 48
Input/Output (I/O) 2
Insert mode (EDLIN) . 32, 33
INSTALL command .. 41, 75
Internal commands 6
IO.SYS 3

JOIN command 75

KEYB command 75
KEYBOARD.SYS 43
KEYB UK command 43

LABEL command 75
LASTDRIVE 41
Line editor 31
List mode (EDLIN) 32
Listing directories 25

MD command 6, 23, 75
MEM command 75
MODE command 75
MORE filter command 55, 77
MOUSE.SYS 40
Move mode (EDLIN) 34
MSDOS.SYS 3

NLSFUNC command 77
NOW command 77

Option 9
Options with
  BACKUP 56
  CHKDSK 19
  COMMAND.COM 52
  COPY 16
  DEVICE 40
  DIR 9
  FORMAT 13
  PRINT 56, 78
  RESTORE 56
  XCOPY 20

PATH command ... 6, 43, 77
PAUSE command 48
PRINT command 56, 78
PRTSC command 56
PRN re-direction 54
Prompt 4
PROMPT com'nd 6, 23, 44, 78

Quit EDLIN 36

RAM 5
RD command 6, 23, 78

RECOVER command 78
Re-direction 54
REM command 42,49
RENname command 6, 18, 78
REPLACE command 79
Replace mode (EDLIN) ... 35
RESTORE command 56, 79
ROM 5
Root directory 21

Search mode (EDLIN) 34
SELECT command .. 60, 79
SET command 6, 53, 79
SHARE command ... 59, 80
SHELL 42
SHIFT command 49
Size of discs 14
SORT filter command 55, 80
Sub-directories 21
SUBST command 80
Switch 9
SWITCHES command 80
SYS command 15, 80
System disc 3

TIME command ... 6, 53, 81
Transfer mode (EDLIN) ... 36
TREE command 26, 81
Type of discs 14
TYPE command 53, 81
Text files 31

Utility disc 4

VDISK command 40, 81
VER command 6, 54, 81
VERIFY command 6, 81
VOL command 6, 54, 81

Wildcard characters 10

XCOPY command ... 20, 81

NOTES

NOTES

BERNARD BABANI BP232

A Concise Introduction to MS-DOS

If you are a PC user and want to get the most out of your computer in terms of efficiency and productivity, then you must learn its MS-DOS operating system. With this CONCISE INTRODUCTION TO MS-DOS you will learn to do just that in the shortest and most effective way.

The book was written with the non-expert, busy person, in mind and, as such, it has an underlying structure based on "what you need to know first, appears first". Nonetheless, the book was also written in such a way as to allow the more experienced user to start from any section, as each section was designed to be self-contained.

The book explains:-

How the DOS operating system is structured so that you understand what happens when you first switch on your computer.

How to use the DOS commands to perform various house-keeping operations on your disc files.

How to use the EDLIN line editor to fully configure your system by writing your own CONFIG.SYS and AUTOEXEC.BAT files.

How directories and sub-directories can be employed to organize the way you keep files on your disc so that your system's efficiency is maximized.

How to write batch files to automate the operation of your system.

How to manage your system's environment.

The book covers all versions of 3.x and 4.x of both PC-DOS and MS-DOS as implemented by IBM and Microsoft for other manufactures of "compatible" PCs, with coverage extending to both the hard and floppy disc-based systems. A summary of all DOS commands is given, with examples, in the penultimate chapter of the book, which turns it into a useful reference guide. A separate section of the book deals with the DOS shell of version 4.x, which is a menu-driven graphical interface making this version of the operating system easier to use.

ISBN 0-85934-177-1

£2.95

00295

9 780859 341776